هياكل البيانات

بلغة

C++

محاضرات الأستاذ الفاضل

م.م علي عادل العامري

إعداد الطالب

علاء خضير الزيادي

جامعة البصرة – كلية التربية

قسم علوم الحاسبات 2009 - 2008

الفهرس

1- المقدمة:

هياكل البيانات Data Structure:

البيانات Data:

هي مجموعة من الحقائق والافكار التي قد تكون حروفا او ارقاما او صورا او مقاطع فيديو او خليطا من هذه الامور.

المعلومات Information:

هي مجموعة من الحقائق والافكار عن شئ تمت معالجته.

انواع المعالجة Processing Type:

1- الاضافة.

2- الحذف.

3- الدمج.

4- الفرز والترتيب.

5- التحليل والتركيب باستخدام التعابير الحسابية.

6- النسخ الالكتروني.

7- الحماية والفك.

8- الاسترجاع والتعديل.

التركيب الفيزيائي والمنطقي للبيانات:

Physical Structure And Logical Structure of Data:

التركيب الفيزيائي الحقيقي:

وهو كيفية تركيب البيانات على الاوساط التخزينية من القرص الصلب والقرص المرن والقرص المغناطيسي ، حيث تخزن البيانات اما بشكل مباشر او بطريقة تتابعية.

التركيب المنطقي:

هو ترتيب معلومات البرنامج بشكل معين حيث يتم تنفيذ هذا البرنامج بطريقة صحيحة.

هياكل البيانات Data Structure :

عبارة عن آليات وخوارزميات معينة لوظع برامج بحيث تطبق بشكل جيد ، ونستفاد منها في فهم كيفية برمجة قواعد البيانات ، واستغلال مواقع الذاكرة بشكل ومنظم ، وبالتالي اعطائها تعريف اكثر دقة بكونها طريقة ترابط وتنظيم البيانات مع بعظها

البعض في الذاكرة ، بحيث تأخذ شكلا و هيكلا معينا في تلك الذاكرة فتعتبر بنية عضوية لمجموعة من عناصر البيانات المتطابقة نوعا وشكلا والتي تنظم في نسق واحد لتؤدي غرضا محددا.

فوائد هياكل البيانات **Advantages Of Data Structure:**

1- التحكم في توزيع البيانات والتعرف الى طبيعتها وبنائها الاساسي بنسق معين في الذاكرة.

2- تمكين المبرمج من ابداع طرق مبتكرة لكتابة البرامج المختلفة.

3- اختصار زمن التخزين واسترجاع البيانات من الذاكرة.

4- بناء برامج قوية ومتماسكة من حيث البناء والمنطق .

انواع هياكل البيانات :

1- هياكل بيانات ثابتة:

كالمتجهات والجداول والسجلات ، حيث عند الاعلان عنها يجب تحديد حجم هذه البيانات فلا تقبل الاظافة فوق حجمها المحدد..

2- هياكل بيانات متحركة (ديناميكية): وتقسم الى نوعين:

أ) هياكل بيانات خطية متغيرة: كالقوائم والطوابير والملفات والمكدسات والابجديات والمجموعات..

ب) هياكل بيانات متشعبة (عشوائية): كالاشجار والخرائط..

2- المصفوفات **The Arrays:**

3- المصفوفة الأحادية البعد **One Dimension Arrays :**

عبارة عن هيكل بياني يتكون من مجموعة من العناصر والتي جميعها من نفس النوع البياني ، تتميز بكون حجمها ثابت لا يتغير اثناء الاعلان عنها.

الاعلان عن مصفوفة:

Ex:

```
int A[10];
float A[20];
```

طريقة خزن المصفوفة داخل الذاكرة:

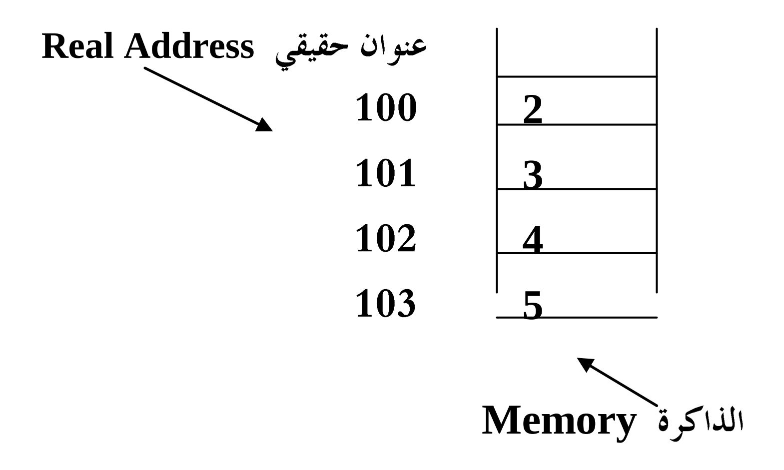
Ex:

A[4]

A=[2 3 4 5]

A[0]=2 , A[1]=3 , A[3]=4 , A[4]=5

عناوين نسبية وليست حقيقية



معادلة الوصول لكل عنصر :

$$Loc\ (\ A\ [\ I\ ]\ ) = Real\ Address + I$$

مثال:

$$Loc\ (\ A\ [\ 3\ ]\ ) = 100 + 3 = 103 = 5$$

عملية الخزن في الذاكرة :

1- تحتاج الى حجز كتلة كاملة بمقدار حجم المصفوفة.

2- طريقة الخزن بشكل متسلسل.

3- طريقة الوصول بشكل عشوائي ، بالاعتماد على معادلة الوصول.

الامثلة:

1- برنامج لقراءة مصفوفة احادية ذات **N** عنصر من النوع الصحيح ثم ينشأ مصفوفة جديدة تضم العناصر المتكونة من مرتبتين في المصفوفة الاولى؟

```
#include<iostream.h>
void main()
{
 int n,a[100],b[100];
cout<<"Enter n"<<endl;
cin>>n;
```

```
int k=0;
for(int i=0;i<n;i++)
{
 cout<<"Enter Array :"<<endl;
cin>>a[i];}
for(i=0;i<n;i++)
if((a[i]>=10)&&(a[i]<=99))
{ b[k]=a[i];
k++;}
for(int j=0;j<k;j++)
cout<<b[j]<<" ";
cin>>"";
 }
```

2- برنامج لقراءة مصفوفة احادية ذات N عنصر من النوع الصحيح ويحسب:

أ) انشاء مصفوفة جديدة تمثل استبدال النصف الاول من المصفوفة محل النصف الثاني؟

ب) حساب عدد العناصر الغير متكررة في المصفوفة؟

```
#include<iostream.h>
const int size=100;
void readID(int[],int);
void swap(int[],int);
void count(int[],int);
void printID(int[],int);
void main()
{
 int n,a[size],k;
cout<<"Enter n"<<endl;
cin>>n;
 readID(a,n);
 swap(a,n);
 count(a,n);
}
void readID(int a[size],int n)
{
for(int i=0;i<n;i++)
{
  cout<<"Enter Array :"<<endl;
  cin>>a[i];}
}
void swap(int a[size],int n)
{ int x;
```

```
  if(n%2==0)
  x=n/2;
else
x=n/2+1;
int h=x;
for(int i=0;i<x;i++)
{
 int z=a[i];
a[i]=a[h];
a[h]=z;
h++;}
 printID(a,n);
}
void printID(int a[size],int n)
{
  for(int i=0;i<n;i++)
  cout<<a[i]<<endl;
}
void count(int a[size],int n)
{ int f=0;
  for(int i=0;i<n;i++)
{
 int item=a[i];
  for(int j=i+1;j<n;j++)
{
 if(item!=a[j])
f++;
}
cout<<f<<endl;
cin>>"";
}
}
```

3- برنامج لقراءة مصفوفة احادية ذات N عنصر من النوع الرمز الواحد ثم ينشأ مصفوفة جديدة تضم العناصر التي هي عبارة عن عمليات حسابية في المصفوفة الاولى؟

```
#include<iostream.h>
const int size=100;
void readID(char[],int);
void process(char [],int, char [],int&);
void main()
{
```

```
 char a[size],b[size];
int n,k;
cout<<"Enter n"<<endl;
cin>>n;
 readID(a,n);
 process(a,n,b,k);
}
void readID(char a[size],int n)
{
for(int i=0;i<n;i++)
{
  cout<<"Enter Array :"<<endl;
  cin>>a[i];}
}
void process(char a[size],int n, char b[size],int &k)
{
k=0;
for(int i=0;i<n;i++)
{
 if((a[i]=='*')||(a[i]=='+')||(a[i]=='/')||(a[i]=='-'))
{
 b[k]=a[i];
 k++;
}
}
for(i=0;i<n;i++)
cout<<b[i]<<endl;
cin>>"";
}
```

4– برنامج لقراءة مصفوفة احادية ذات N عنصر من النوع الصحيح ثم ينشأ مصفوفة جديدة تضم العناصر الموجبة والتي هي من مضاعفات الاربعة والستة في المصفوفة الاولى؟

```
#include<iostream.h>
const int size=100;
void readID(int[],int);
void process(int [],int, int [],int&);
void printID(int[],int);
void main()
{
 int a[size],b[size];
```

```
int n,k;
cout<<"Enter n"<<endl;
cin>>n;
 readID(a,n);
 process(a,n,b,k);
printID(b,n);
}
void readID(int a[size],int n)
{
for(int i=0;i<n;i++)
{
  cout<<"Enter Array :"<<endl;
  cin>>a[i];}
}
void process(int a[size],int n, int b[size],int &k)
{
k=0;
for(int i=0;i<n;i++)
{
 if((a[i]>0)&&(a[i]%6==0)&&(a[i]%4==0))
{
 b[k]=a[i];
 k++;}
}
}
void printID(int a[size],int n)
{
for(int i=0;i<n;i++)
cout<<a[i]<<endl;
cin>>"";
}
```

5– برنامج لقراءة مصفوفة احادية ذات N عنصر من النوع الصحيح ثم ادخل عنصر معين واحسب عدد مرات تكرار ذلك العنصر؟

```
#include<iostream.h>
const int size=100;
void readID(int[],int);
void rep(int [],int,int);
void main()
{
```

```
 int a[size];
int n,x;
cout<<"Enter n"<<endl;
cin>>n;
 readID(a,n);
 rep(a,n,x);
}
void readID(int a[size],int n)
{
for(int i=0;i<n;i++)
{
  cout<<"Enter Array :"<<endl;
  cin>>a[i];}
}
void rep(int a[size],int n,int x)
{
int i,c=0;
cout<<"Enter X "<<endl;cin>>x;
for(i=0;i<n;i++)
{
 if(a[i]==x)
c++;}
cout<<c;
cin>>"";
}
```

6- برنامج لقراءة مصفوفة احادية ذات N عنصر من النوع الصحيح ثم ادخل عنصر معين واستبدله اينما تجده في المصفوفة المقروءة؟

```
#include<iostream.h>
const int size=100;
void readID(int[],int);
void swap(int[],int,int&);
void printID(int[],int);
void main()
{
 int n,a[size],x;
cout<<"Enter n"<<endl;
cin>>n;
 readID(a,n);
 swap(a,n,x);
```

```
 printID(a,n);
}
void readID(int a[size],int n)
{
for(int i=0;i<n;i++)
{
  cout<<"Enter Array :"<<endl;
  cin>>a[i];}
}
void swap(int a[size],int n,int& x)
{cout<<"Enter X"<<endl;
cin>>x;
for(int i=0;i<n;i++)
{
 if(x==a[i])
 a[i]=5;
}
}
void printID(int a[size],int n)
{
  for(int i=0;i<n;i++)
  cout<<a[i]<<endl;
  cin>>"";
}
```

7- برنامج لقراءة مصفوفة احادية ذات N عنصر من النوع الصحيح ويحسب:

أ) معدل العناصر الزوجية الموجبة في تلك المصفوفة؟

ب) حاصل ضرب عناصر تلك المصفوفة؟

```
#include<iostream.h>
const int size=100;
void readID(int[],int);
void avpe(int[],int);
void prod(int[],int);
void main()
{
 int n,a[size];
cout<<"Enter n"<<endl;
cin>>n;
 readID(a,n);
 avpe(a,n);
```

```
 prod(a,n);
}
void readID(int a[size],int n)
{
for(int i=0;i<n;i++)
{
  cout<<"Enter Array :"<<endl;
  cin>>a[i];}
}
void avpe(int a[size],int n)
{
  int i,s=0,k=0;
  for(i=0;i<n;i++)
{
 if((a[i]>0)&&(a[i]%2==0))
 {k++;
 s+=a[i];}
}
  float ave=(s/k);
  cout<<ave<<endl;
}
void prod(int a[size],int n)
{ int f=1;
  for(int i=0;i<n;i++)
  f*=a[i];
  cout<<f<<endl;
  cin>>"";
}
```

8- برنامج يقرأ مصفوفة احادية ويطبعها بالمقلوب؟

```
#include<iostream.h>
#include<conio.h>
void main()
{clrscr();
int i,a[100],n;
cout<<"Enter size of array: "<<endl;
cin>>n;
for(i=0;i<n;i++)
{cout<<"Enter the elements: ";
cin>>a[i];}
for(i=n-1;i>=0;i--)
cout<<a[i]<<"  ";
```

```
cin>>" ";
}
```

10– مشروع برنامج يضم عمليات الحذف – الإضافة (بداية ، وسط ، نهاية) والبحث على مصفوفة أحادية البعد؟

```
#include<iostream.h>
#include<conio.h>
void Read1D(int[]);
void Print1D(int[]);
int Search(int[],int);
void DeleteFirst(int[]);
void DeleteLast(int[]);
void DeleteMid(int[]);
void InsertFirst(int[],int);
void InsertLast(int[],int);
void InsertMid(int[],int);
int size=0;
int totalsize=10;
void main()
{
 int ch,a[100],item,loc,ch2;
 do
 {
    cout<<"1-Read1D................."<<endl;
    cout<<"2-Print1D................"<<endl;
    cout<<"3-Search................."<<endl;
    cout<<"4-Delete................."<<endl;
    cout<<"5-Insert................."<<endl;
    cout<<"0-Exit..................."<<endl;
    cout<<"Enter your choise please:"<<endl;
    cin>>ch;
  if(ch==1)
   Read1D(a);
  else if(ch==2)
   Print1D(a);
  else if(ch==3)
  {
   cout<<"Enter item to search for ?"<<endl;
   cin>>item;
```

```
loc=Search(a,item);
if(loc==-1)
 cout<<"Not Found"<<endl;
else
 cout<<"Found in  : "<<loc<<endl;
}
else if(ch==4)
{
 cout<<"Enter 1: Delete First..."<<endl;
 cout<<"Enter 2: Delete Mid......"<<endl;
 cout<<"Enter 3: Delete Last....."<<endl;
 cout<<" Now enter your choise..."<<endl;
 cin>>ch2;
if(ch2==1)
{
 DeleteFirst(a);
 Print1D(a);
}
else if(ch2==2)
{
 DeleteMid(a);
 Print1D(a);
}
else
{
 DeleteLast(a);
 Print1D(a);
}
 ch2=0;
}
else if(ch==5)
{
 cout<<"1: Insert in the First"<<endl;
 cout<<"2: Insert in the Mid "<<endl;
 cout<<"3: Insert in the Last"<<endl;
 cout<<" Enter your choise   "<<endl;
 cin>>ch2;
if(ch2==1)
{
 cout<<"Enter item :"<<endl;
 cin>>item;
```

```
  InsertFirst(a,item);
  Print1D(a);
 }
 else if(ch2==2)
 {
  cout<<"Enter item :"<<endl;
  cin>>item;
  InsertMid(a,item);
  Print1D(a);}
 else if(ch2==3)
 {
  cout<<"Enter item :"<<endl;
  cin>>item;
  InsertLast(a,item);
  Print1D(a);}
  ch2=0;
 }
 }
while(ch!=0);
  clrscr();
 }
void Read1D(int a[])
 {
 int i,n;
  cout<<"Enter the size of array: "<<endl;
  cin>>n;
  for(i=0;i<n;i++)
  {
  cout<<"Enter elemenet: ";
  cin>>a[i];
  size++;
  }
 }
void Print1D(int a[])
{
 int i;
 for(i=0;i<size;i++)
 cout<<"    "<<a[i];
 cout<<endl;
}
int Search(int a[],int item)
```

```
{
  int i,loc=-1;
  for(i=0;i<size;i++)
  if(a[i]==item)
  loc=i;
  return loc;
}
void DeleteFirst(int a[])
{
 int i;
 if(size>0)
 {
 for(i=0;i<size;i++)
 a[i]=a[i+1];
 size--;}
 else
 cout<<"The Array is empty"<<endl;
 }
void DeleteLast(int a[])
{
 if(size>0)
 size--;
 else
 cout<<"The Array is empty"<<endl;
}
void DeleteMid(int a[])
{
 int i,loc,item;
 cout<<"Enter element: ";
 cin>>item;
loc=Search(a,item);
 if((loc!=0)&&(loc!=size-1)&&(loc!=-1))
 {
  for(i=loc;i<size;i++)
  a[i]=a[i+1];
  size--;
 }
  else
  cout<<"Sory you can't delete? "<<endl;
 }
void InsertFirst(int a[],int item)
```

```
 {
  int i;
  if (size<totalsize)
  {
   for(i=size;i>0;i--)
   a[i]=a[i-1];
   a[0]=item;
   size++;
  }
  else
   cout<<"The Array is full ";
  }
void InsertLast(int a[],int item)
 {
  if(size<totalsize)
  {
   a[size]=item;
   size++;
  }
  else
   cout<<"The Array is full ";
  }
void InsertMid(int a[],int item)
 {
  int i,loc,elm;
  if(size<totalsize)
  {
  loc=Search(a,item);
  if((loc!=0)&&(loc!=size-1)&&(loc!=-1))
  {
  cout<<"Enter element to insert after: "<<endl;
  cin>>elm;
  for(i=size;i>loc;i--)
  a[i]=a[i-1];
  a[loc+1]=elm;
  size++;
  }
  else
  cout<<"Sory you can't insert !"<<endl;}
  else
  cout<<"The Array is full ";
```

}

4- المصفوفة الثنائية البعد   Tow Dimension Arrays:

عبارة عن هيكل بياني يتكون من مجموعة من العناصر والتي جميعها من نفس النوع البياني ، تتميز بكون عناصرها تتوزع على مجموعة من الصفوف والاعمدة التي تمثل حجم تلك المصفوفة.

الاعلان عن مصفوفة:

Ex:

```
Int A[10][10];
Float A[20][20];
```

طريقة خزن المصفوفة داخل الذاكرة:

Ex:

A[2][2]

A=
$$\begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$$

A[0][0]=2 , A[0][1]=3 , A[1][0]=5 , A[1][1]=6

عناوين نسبية وليست حقيقية

طرق الوصول لأي عنصر   Access Methods of any item:

تعتمد طريقة الوصول لأي عنصر من عناصر المصفوفة على طريقة تخزين تلك العناصر ، وهي على نوعين:

1- طريقة الخزن بشكل (صف – صف).

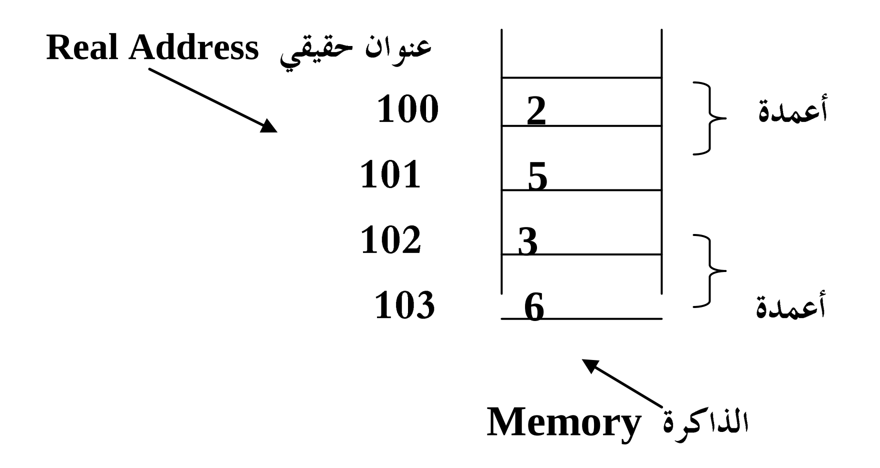
2- طريقة الخزن بشكل (عمود – عمود).

التمثيل المنطقي للمصفوفة الثنائية: **Logical Structure**

$$A = \begin{pmatrix} a00 & a01 & \dots & a0m \\ a10 & a11 & \dots & a1m \\ \vdots & \vdots & & \vdots \\ an0 & an1 & \dots & anm \end{pmatrix} \quad n*m$$

التمثيل الحقيقي للمصفوفة الثنائية: **Physical Structure**

(عمود – عمود):



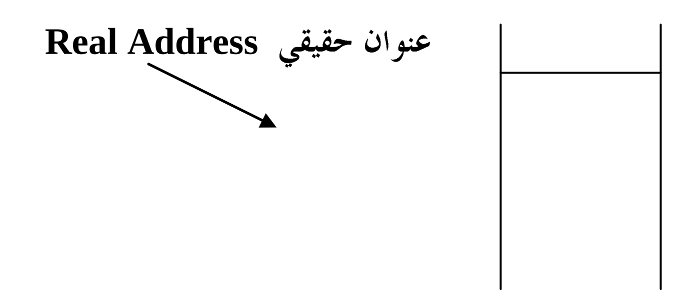
معادلة الوصول:

Loc (A [i][j]) = Real Address + j*m+i

مثال:

Loc (A [0][0]) = 100 + 0 *2+0= 100 = 2

(صف – صف):



100	2	صفوف
101	3	
102	5	صفوف
103	6	

Memory الذاكرة

معادلة الوصول:

Loc (A [i][j]) = Real Address + i*n+j

مثال:

Loc (A [1][0]) = 100 + 1 *2+0= 102 = 5

الأمثلة:

1– برنامج لقراءة مصفوفة ثنائية البعد ومن النوع الصحيح ويحسب:

أ) عدد العناصر الزوجية الموجبة في تلك المصفوفة؟

ب) مجموع عناصر تلك المصفوفة؟

```
#include<iostream.h>
const int size=100;
void read2D(int[size][size],int,int);
void sum(int[size][size],int,int,int&);
void countPE(int[size][size],int,int);
void main()
{
 int n,m,s,a[size][size];
cout<<"Enter N , M"<<endl;
cin>>n>>m;
 read2D(a,n,m);
 sum(a,n,m,s);
 cout<<s<<endl;
 countPE(a,n,m);
}
void read2D(int a[size][size],int n,int m)
{
for(int i=0;i<n;i++)
 for(int j=0;j<m;j++)
```

```
{
 cout<<"Enter Array 2D :"<<endl;
 cin>>a[i][j];}
}
void sum(int a[size][size],int n,int m,int& s)
{
 int i,j;
 s=0;
 for(i=0;i<n;i++)
 for(j=0;j<m;j++)
   s=s+a[i][j];
 }
void countPE(int a[size][size],int n,int m)
{ int c=0;
  for(int i=0;i<n;i++)
  for(int j=0;j<m;j++)
  {
   if((a[i][j]%2==0)&&(a[i][j]>0))
   c++;}
  cout<<c<<endl;
  cin>>"";
 }
```

تمارين:

2- برنامج لقراءة مصفوفة ثنائية البعد **N*M** من النوع الصحيح ثم انشأ مصفوفة جديدة تمثل مضروب كل عنصر في المصفوفة المقروءة؟

3- برنامج لقراءة مصفوفة ثنائية البعد **N*M** من النوع الصحيح ثم احسب التالي:

أ) **B=X*A**

ب) **C=A+B**

4- برنامج لقراءة مصفوفة ثنائية البعد **N*M** من النوع الحقيقي ثم ولد مصفوفتين احاديتين ؛ الاولى: تضم الاجزاء الصحيحة ، والثانية: تضم الاجزاء الحقيقية. من المصفوفة المقروءة؟

5- مشروع برنامج يضم عمليات الحذف – الإضافة (بداية ، وسط ، نهاية) والبحث على مصفوفة ثنائية البعد؟

```
#include<iostream.h>
#include<conio.h>
void Read2D(int[10][10],int&,int&);
void Print2D(int[10][10],int&,int&);
```

```
void Search(int[10][10],int&,int&,int);
void DeleteFirst(int[10][10],int&,int&);
void InsertFirst(int[10][10],int&,int&);
int locr,locc;
void main()
{
 clrscr();
 int a[10][10],n,m,item,ch;
  do
  {
     cout<<"Enter 1:for Read2D:"<<endl;
     cout<<"Enter 2:for print2D:"<<endl;
     cout<<"Enter 3:for search:"<<endl;
     cout<<"Enter 4:for delete first:"<<endl;
     cout<<"Enter 5:for Insert first"<<endl;
     cout<<"Enter 0:to stop.";
     cin>>ch;
     if(ch==1)
     Read2D(a,n,m);
      if(ch==2)
  Print2D(a,n,m);
   if(ch==3)
  {
   cout<<"Enter item to search for ?"<<endl;
   cin>>item;
  Search(a,n,m,item); }
 else if(ch==4)


  DeleteFirst(a,n,m);
  else if(ch==5)


  InsertFirst(a,n,m);
  }
     while(ch!=0);
  }
void Read2D(int a[10][10],int& n,int& m)
  {
  int i,j;
   cout<<"Enter the size of array: "<<endl;
   cin>>n>>m;
```

```
  for(i=0;i<n;i++)
  { for(j=0;j<m;j++)

  {cout<<"Enter elemenet: ";
  cin>>a[i][j];}
  cout<<endl;
  }
 }
 void Print2D(int a[10][10],int& n,int& m)
{
 int i,j,k;
 for(i=0;i<n;i++)
 {for(j=0;j<m;j++)
 cout<<"  "<<a[i][j];
 cout<<endl;}
}
void Search(int a[10][10],int& n,int& m,int item)
{
  int i,j,f=0;locr=-1;locc=-1;
 for(i=0;i<n;i++)
 for(j=0;j<m;j++)
 {
 if(a[i][j]==item)
 {f=1;
 locr=i;
 locc=j;
 }}
 if(f==0)
  cout<<"Not Found"<<endl;
  else
 cout<<"Found in :"<<"["<<locr<<"]"<<"["<<locc<<"]"<<endl;
 }
void DeleteFirst(int a[10][10],int& n,int& m)
{
 int i,j;
 for(i=0;i<n;i++)
for(j=0;j<m;j++)
 {
 if(j==m-1)
 a[i][j]=a[i+1][m-j-1];
  else
```

```
 a[i][j]=a[i][j+1];
 }
a[n-1][m-1]=0;

Print2D(a,n,m);

}

  void InsertFirst(int a[10][10],int& n,int& m)
{
 int i,j,item,k1,k2;
 cout<<"Enter item"<<endl;
 cin>>item;
 k1=n;
 k2=0;
 for(i=n;i>=0;i--)
 for(j=m-1;j>=0;j--)
 {
 a[k1][k2]=a[i][j];
 k1=i;
 k2=j;
 }
 a[0][0]=item;
 n++;
Print2D(a,n,m);
}
```

5– الخيوط الرمزية :Strings

عبارة عن هيكل بياني يتكون من مجموعة من العناصر التي هي عبارة عن رموز تتمثل بالحروف الأبجدية والأرقام والرموز الخاصة التي تتواجد على لوحة المفاتيح ويعلن عنها بالطريقة التالية:

```
1- char  name - of - string[size];
2- char  *name;
```

Ex:-

```
    1)   char   name[35];
    2)   char  *name;
```

*ابرز الدوال التي تطبق على الخيوط الرمزية والتي تتواجد ضمن المكتبة <string.h> :

1- **strcpy(st1,st2);** : دالة لاستنساخ خيط رمزي معين من خيط آخر ، ومكن تحديد عدد الرموز المراد استقطاعها وبالشكل التالي: **strncpy(st1,st2,n);** تمثل **n** عدد الرموز المستقطعة.

2- **strcmp(st1,st2);** : دالة للمقارنة بين الخيوط الرمزية.

3- **strlen(st);** : دالة لحساب طول الخيط الرمزي .

4- **strcat(st1,st2);** : دالة لدمج خيطين رمزيين بحيث تكون النتيجة في الخيط الرمزي الاول.وممكن تحديد عدد الرموز التي تستقطع من الخيط الرمزي الثاني وتدمج الخيط الاول بواسطة: **strncat(st1,st2,n);** .

**ابرز الدوال التي تطبق على الخيوط الرمزية والتي تتواجد ضمن المكتبة <ctype.h> :

1- **isalnum(ch);** : دالة لمعرفة الرمز اذا كان عبارة عن حرف ابجدي أو رقم ، ترجع هذه الدالة قيمة صفرية اذا كان الرمز لا يساوي قيمة ابجدية ولا رقم.

2- **isalpha(ch);** : تختبر الرمز اذا كان حرف ابجدي او لا.

3- **islower(ch);** : تختبر الرمز فيما اذا كان صغير ترجع قيمة والا ترجع صفر.

4- **isupper(ch);** : تختبر الرمز اذاكان كبير فانها ترجع قيمة.

5- **isdigit(ch);** : تختبره اذا كان رقم ترجع قيمة.

6- **struper();** : تحول الحرف من صغير الى كبير.

7- **strlwr();** : تحول الحرف من كبير الى صغير.

8- **strinv();** : تعكس الخيط الرمزي.

الامثلة:

1- برنامج لقراءة خيط رمزي ثم حساب عدد الرموز التي عبارة عن ارقام؟

```
#include<iostream.h>
#include<string.h>
void main()
{char st[100],l;
cout<<"Enter st: "<<endl;
cin>>st;
l=strlen(st);
int c=0;
for(int i=0;i<l;i++)
if((st[i]>='0')&&(st[i]<='9'))
c++;
cout<<c;
}
```

2- برنامج لقراءة خيط رمزي ثم طباعة الحرف الذي يكون في المنتصف؟

```
#include<iostream.h>
#include<string.h>
void main()
{int i,l;
char st[30];
cout<<"enter string: ";
cin>>st;
l=strlen(st);
i=l/2;
cout<<st[i];
}
```

3- برنامج لقراءة خيط رمزي ثم تحويل الحرف الكبير الى صغير؟

```
#include<iostream.h>
#include<string.h>
#include<ctype.h>
void main()
{
 int i,l;
 char st[10];
 cout<<"enter your string: "<<endl;
 cin>>st;
 l=strlen(st);
 for(i=0;i<l;i++)
 {
  if(isupper(st[i])!=0)
  strlwr(st);
  }
   for(i=0;i<l;i++)
 cout<<st[i];
 cin>>" ";
}
```

4- برنامج لقراءة خيط رمزي ثم تحويل الحرف الصغير الى كبير؟

```
#include<iostream.h>
#include<string.h>
#include<ctype.h>
void main()
{
```

```
 int i,l;
 char st[10];
 cout<<"enter your string: "<<endl;
 cin>>st;
 l=strlen(st);
 for(i=0;i<l;i++)
 {
  if(islower(st[i])!=0)
  strupr(st);
  }
   for(i=0;i<l;i++)
 cout<<st[i];
 cin>>" ";
}
```

5- برنامج لقراءة خيط رمزي ثم تحويل كل رقم الى الرمز * ؟

```
#include<iostream.h>
#include<string.h>
#include<ctype.h>
void main()
{
 int i,l;
 char st[10];
 cout<<"enter your string: "<<endl;
 cin>>st;
 l=strlen(st);
 for(i=0;i<l;i++)
 if(isdigit(st[i])!=0)
  st[i]='*';
   for(i=0;i<l;i++)
 cout<<st[i];
 cin>>" ";
}
```

6- برنامج لقراءة خيط رمزي ثم ادخال رمز معين والبحث عنه اذا كان موجود في الخيط الرمزي الاول او لا ؟

```
#include<iostream.h>
#include<string.h>
void main()
{
```

```
int i,l,f=0;
char st[100],c;
cout<<"Enter st: "<<endl;
cin>>st;
l=strlen(st);
cout<<"Enter the char for search in the string: "<<endl;
cin>>c;
for(i=0;i<l;i++)
{if(st[i]==c)
f=1;
}
if(f==0)
cout<<"not found ";
else
cout<<"char is found in the string "<<f;
}
```

7- برنامج لقراءة خيط رمزي ثم حساب التالي:

أ) عدد الرموز التي هي عبارة عن حروف ابجدية صغيرة؟

ب) عدد الرموز التي هي عبارة عن ارقام؟

```
#include<iostream.h>
#include<string.h>
#include<ctype.h>
void main()
{
 int i,l,cd=0,cl=0;
 char st[100];
 cout<<"enter your string: "<<endl;
 cin>>st;
 l=strlen(st);
 for(i=0;i<l;i++)
 {
  if (isdigit(st[i])!=0)
  cd++;
  else if(islower(st[i])!=0)
  cl++;
  }
```

```
 cout<<"result "<<cl<<"::"<<cd<<endl;
}
```

6- الهياكل ((القيود)) Structure or Recorders:

وهي عبارة عن هيكل بياني يتكون من مجموعة من الانواع البيانية المختلفة والتي يعبر عنها بالحقول ، بالاضافة الى ان كل هذه الحقول يطلق عليها بالقيد.

الاعلان عنها كالتالي:

```
struct  struct - name
 {
   Data type1   field 1;
   Data type2   field 2;
  };
```

Ex:-

```
      struct  student
       {
         char  name[30];
         int     no;
         char  address[30];
        };
```

الامثلة:

1- برنامج لقراءة قيود N من الطلبة ، كل قيد يتكون من الاسم والمعدل ، احسب عدد الطلبة الناجحين؟ ثم اطبع بيانات الطلبة؟

```
#include<iostream.h>
#include<string.h>
const int size=100;
struct student
```

```
{
char name[35];
int avg;
};
void read(student[],int);
void print(student[],int);
void main()
{
int i,n;
student s[size];
cout<<"Enter N "<<endl;
cin>>n;
read(s,n);
print(s,n);
}
void read(student s[size],int n)
{
int i,c=0;
for(i=0;i<n;i++)
{
cout<<"Enter in formation "<<endl;
cin>>s[i].name>>s[i].avg;
if(s[i].avg>=50)
c++;
}
}
void print(student s[size],int n)
{
int i,c;
for(i=0;i<n;i++)
cout<<s[i].name<<"::"<<s[i].avg<<endl;
cout<<"the sumation succeses :"<<c;
cin>>" ";
}
```

2- برنامج لقراءة قيود N من الطلبة ، كل قيد يتكون من الاسم والكلية والقسم ، أطبع أسماء الطلبة الذين هم في كلية التربية وفي قسم الحاسبات أو الرياضيات؟

```
#include<iostream.h>
#include<string.h>
```

```
const int size=100;
struct student
{
char name[35],col[35],dep[35];
};
void readrec(student[],int);
void print(student[],int);
void main()
{
int i,n;
student s[size];
cout<<"Enter N :";
cin>>n;
readrec(s,n);
print(s,n);
}
void readrec(student s[size],int n)
{ int i;
for(i=0;i<n;i++)
{
cout<<"Enter The Informaition: "<<endl;
cin>>s[i].name>>s[i].col>>s[i].dep;
}
}
void print(student s[size],int n)
{
int i;
for(i=0;i<n;i++)
{
if((strcmp(s[i].col,"edu")==0)&&(strcmp(s[i].dep,"comp")==0)||(strcmp(s[i].d
ep,"math")==0))
cout<<s[i].name<<endl;}
cin>>" ";
}
```

3- برنامج لقراءة قيود N من الموظفين ، كل قيد يتكون من الاسم والدرجة الوظيفية والراتب والعنوان والشهادة وسنة التعيين ، بحيث تتوزع الرواتب بالاعتماد على الشهادة وبالشكل التالي :-

creat	salary
Phd	1000$

Msc	750$
Bsc	500$
DB	350$
SC	200$

ثم احسب التالي:

أ) اطبع بيانات الموظفين الذين رواتبهم بين (500-1000) ؟

ب) عدل على رواتب الموظفين الذين مضى على عمرهم الوظيفي اكثر من **5** سنوات باجراء زيادة على رواتبهم بقدار (**0.05**) ؟

ج) احسب عدد الموظفين الذين هم من سكنة محافظة البصرة ومن حملة شهادة الدكتوراة **(Phd)** ؟

```
#include<iostream.h>
#include<string.h>
const int size=100;
 struct emp
 {
  char name[35],cer[35],add[35];
  int deg,yearemp;
  float salary;
  };
  void readrec(emp[],int);
  void printrec(emp[],int);
  void editrec(emp[],int);
  void countrec(emp[],int);
  void main()
  { int n;
    emp e[size];
    cout<<"Enter n :"<<endl;
    cin>>n;
   readrec(e,n);
   printrec(e,n);
   editrec(e,n);
   countrec(e,n);
  }
  void readrec(emp e[size],int n)
  {
   for(int i=0;i<n;i++)
   {
    cout<<"Enter informaiton: "<<endl;
```

```
  cin>>e[i].name>>e[i].deg>>e[i].cer>>e[i].add>>e[i].yearemp;
 }
  if(strcmp(e[i].cer,"phd")==0)
  e[i].salary=1000;
  else if(strcmp(e[i].cer,"msc")==0)
  e[i].salary=750;
  else if(strcmp(e[i].cer,"bsc")==0)
  e[i].salary=500;
  else if(strcmp(e[i].cer,"db")==0)
  e[i].salary=350;
  else if(strcmp(e[i].cer,"sc")==0)
  e[i].salary=200;
 }
void printrec(emp e[size],int n)
 {
  for(int i=0;i<n;i++)
  {
   if((e[i].salary>=500)&&(e[i].salary<=1000))

cout<<e[i].name<<e[i].deg<<e[i].salary<<e[i].cer<<e[i].add<<e[i].yearemp<<en
dl;
   }
  }
void editrec(emp e[size],int n)
{
  for(int i=0;i<n;i++)
  {
   if((2008-e[i].yearemp)>5)
   e[i].salary=e[i].salary+(e[i].salary*0.05)
   }
  }
void countrec(emp e[size],int n)
 {
 int i,c=0;
  for(i=0;i<n;i++)
  {
   if((strcmp(e[i].add,"basrah")==0)&&(strcmp(e[i],"phd")==0))
   c++;
  }
  cout<<c;
}
```

4- لديك قائمتين الاولى تمثل مشتركي شركة الاتصالات المتنقلة (الاثير) ، والثانية تمثل مشتركي شركة (آسيا سيل) ، كل قائمة تتكون من مجموعة من القيود وكل قيد يتكون من الاسم والرقم والعنوان والنوع (اذا 2 مشترك موظف ، 1 مشترك عادي) ثم احسب :

أ) عدد الموظفين في كلتا الشركتين؟

ب) اطبع بيانات المشتركين الذين يتواجدون في شركة الاثير وشركة آسيا سيل؟

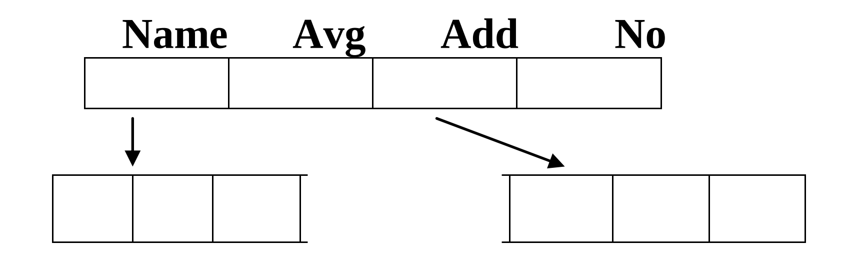
```
#include<iostream.h>
#include<string.h>
#include<conio.h>
const int size=100;
 struct list
 {
  char name[35],add[35];
  int type;
  long no;
  };
  void readrec(list[],int);
  void countrec(list[],int,int &);
  void process(list[],int,list[],int);
  void main()
  {clrscr();
   int n,m,c1,c2;
   list at[size],as[size];
   cout<<"Enter n,m ";
   cin>>n>>m;
   readrec(at,n);
   readrec(as,m);
   countrec(at,n,c1);
   countrec(as,m,c2);
   cout<<c1+c2<<endl;
   process(at,n,as,m);
  }
  void readrec(list r[size],int n)
  {
   for(int i=0;i<n;i++)
   {
    cout<<"Enter informaiton: "<<endl;
    cin>>r[i].name>>r[i].no>>r[i].add>>r[i].type;
```

```
}
}
   void countrec(list r[size],int n,int & c)
{
int i;
c=0;
 for(i=0;i<n;i++)
 if((r[i].type)==2)
 c++;
 }
void process(list r[size],int n,list r1[size],int m)
{
int i,j;
 for(i=0;i<n;i++)
 for(j=0;j<m;j++)

  {if((strcmp(r[i].name,r1[j].name)==0)&&(strcmp(r[i].add,r1[j].add)==0))
  for(i=0;i<n;i++)

 cout<<r[i].name<<r[i].no<<r[i].add<<r[i].type<<endl;
 cin>>"";
 }
 }
```

7– القيود المتداخلة : Nested Structures



طريقة الإعلان :

```
struct  names
 {
    char  first[30],second[30],third[30];
  };
 struct  address
   {
    char  country[30],city[30],street[30];
    };
struct  student
  {
   names  name;
   float  avg;
   address   add;
   int  no;
  };
```

⇩

```
void  main()
{
 student  S;
 S.name.first;
 S.name.second;
 S.avg;
 S.address.city;
 S.address.street; }
```

الامثلة:

1- برنامج لقراءة قيود N من الطلبة ، كل قيد يتكون من الاسم (الاسم الأول ، اسم العائلة) ، والرقم ، والعنوان (المدينة ، الشارع) ، احسب عدد الطلبة الذين تبدأ مدنهم بحرف B ؟

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
const int size=100;
 struct names
 {
```

```
char first[35],family[35];
};
struct address
{
 char city[35],street[35];
 };
 struct student
 {names name;
 address add;
 int no;
 };
void readrec(student[],int);
void countr(student[],int);
void main()
{
  clrscr();
  int n;
 student s[size];
 cout<<"Enter n ";
 cin>>n;
 readrec(s,n);
 countr(s,n);
}
void readrec(student s[size],int n)
{
int i;
for(i=0;i<n;i++)
 {
  cout<<"Enter informaiton: "<<endl;

cin>>s[i].name.first>>s[i].name.family>>s[i].no>>s[i].add.city>>s[i].add.street;
 }
 void countr(student s[size],int n)
 {
 int i,c=0;
 char ch;
  for(i=0;i<n;i++)
  {strncpy(ch,s[i].add.city,1);
  if(strcmp(ch,'b')!=0)
  c++;
  }
```

```
cout<<c;
cin>>"";
}
```

8- المؤشرات : Pointers

هي واحدة من اهم الهياكل البيانية والتي تساعدنا على انشاء هياكل بيانية غير ثابتة الحجم ، وكذلك محاكاة الاستدعاء بالاشارة.

خصائص المؤشرات:

1- متغير يحتوي على عنوان موقع في الذاكرة الى اسم المتغير الذي يشير اليه.

2- المتغير يحتوي على اشارة مباشرة.

3- يمكن للمتغير ان يحتوي على عنوان متغير محمل بقيمة.

الاعلان عن متغير (مؤشر):

Ex:

```
int   x ,*xptr;
float  y ,*yptr;

xptr=0;  (pointer is nothing )
yptr=NULL;  (pointer to nothing )
```

العمليات على المؤشرات:

Ex:

```
int  y=5 , *yptr;
yptr= &y;

cout<<yptr;
```

Notes:

y , * yptr;	y	قيمة
& y , yptr;	y	عنوان

***& yptr   or   &* yptr   (*,&)**   احدهما متمم (يلغي) الآخر

مثال(**1**):

برنامج لحساب المكعب لقيمة صحيحة بواسطة استخدام ثلاث دوال الاولى: تعمل بالاستدعاء بالقيمة **Call by Value**

والثانية: تعمل بواسطة الاستدعاء بمؤشر القيمة **Call by Reference using Reference Pointer**

والثالثة تعمل بواسطة الاستدعاء بالمؤشر **Call by Reference using Pointer** ؟

```
#include<iostream.h>
void cub1(int);
void cub2(int&);
void cub3(int*);
void main()
 {
   int  x;
   cout<<"Enter  x"<<endl;
   cin>>x;
   cub1(x);
   cout<< x<<endl;
   cub2(x);
   cout<< x<<endl;
   cub3(&x);
   cout<< x<<endl;
   cin>>"";
 }
void cub1(int x)
{
  x=x*x*x;}
void cub2(int& x)
{
  x=x*x*x;}
void cub3(int* ptr)
{
  *ptr =*ptr * *ptr * *ptr;
```

}

مثال(2):

برنامج لقراءة عددين X , Y من النوع الصحيح ثم استدعي دالة تعمل بالاستدعاء بالاشارة ((بواسطة استخدام المؤشرات)) لاستبدال تلك العددين احدهما محل الآخر؟

```
#include<iostream.h>
void swap(int *,int *);
void main()
 {
  int x,y;
  cout<<"Enter x , y"<<endl;
  cin>>x>>y;
  swap(&x,&y);
  cout<<x<<y<<endl;
  cin>>"";
  }
void swap(int* x,int* y)
 {
  int z;
  z=*x;
  *x=*y;
  *y=z;
 }
```

9- المؤشرات مع المصفوفات : Pointer and Arrays

ان اسم المصفوفة عبارة عن مؤشر ثابت.

الاعلان عنها:

Ex:

```
    aptr=&a[10];
    cout<<a[3];
    cout<<aptr[3];
```

Notes

تعتبر (A) متغير ثابت لايمكن تحريكه ، يشير دائما الى الموقع الاول ، بينما (aptr) هو مؤشر متغير.

مثال:

برنامج لقراءة وطباعة مصفوفة أحادية من النوع الصحيح باستخدام دالتين للقراءة والطباعة ((باستخدام المؤشرات))؟

```
#include<iostream.h>
void reading(int*,int);
void printing(int*,int);
void main()
{
int a[10],n;
cout<<"Enter n:"<<endl;
cin>>n;
reading(a,n);
printing(a,n);
}
void reading(int*a,int n)
{
for(int i=0;i<n;i++)
{cout<<"Enter A["<<i<<"]=";
cin>>a[i];}
}
void printing(int*x,int n)
{
for(int i=0;i<n;i++)
cout<<*(x+i)<<endl;
cin>>"";
}
```

10– القوائم المرتبطة : Linked Lists

هي عبارة عن هيكل بياني يتكون من مجموعة من الخلايا وكل خلية تتكون من مجموعة من الحقول ذات الانواع البيانية المختلفة فضلا عن وجود حقل يسمى بالمؤشر يستخدم لغرض الربط بين خلايا القائمة المرتبطة .

تتميز القوائم المرتبطة بمجموعة من الخصائص وبالشكل التالي:

1– الخلايا الخزنية متناثرة في الذاكرة.

2– التخصيص ديناميكي.

3– الوصول تتابعي وفي نفس الوقت هناك بطئ في عملية الوصول للعنصر.

4– الحجم المخصص للعنصر الواحد يضم البيانة بالإضافة الى عنصر الارتباط؟

5– تتميز بمرونة عالية لقدرتها على التقلص والاتساع اثناء وقت التنفيذ وحسب الحاجة .

6– تحتاج الى مهارة في التحكم بالحركة وبعناصر الارتباط ((المؤشرات)) وأي ضياع في احد المؤشرات يؤدي الى ضياع جزء او كل القائمة.

انواع القوائم المرتبطة:

1– القائمة الاحادية الارتباط ((تحتاج الى مؤشر واحد)).

2– القائمة الثنائية الارتباط ((تحتاج الى مؤشرين)).

3– القائمة الدائرية الارتباط ((تحتاج الى مؤشر واحد ولكن العنصر الاخير يرتبط مع العنصر الاول)).

11– القوائم الاحادية الارتباط : Single Linked list

الاعلان عنها:

head رأس القائمة

Ali	200

Jon	X

tail ذيل القائمة

200

```
struct  node
 {
  Data   type 1     Value1;
  Data   type 2     Value2;
  node * Pointer ;            مؤشر يحمل مواصفات العقدة
 }
```

Ex:

```
    struct   node
     {
      char  name[30];
      float  avg;
      node* next;
     }
```

```
void main()
 {
  node* head=NULL, *tail=NULL;
```

*الاعلان داخل البرنامج الرئيسي عن رأس **head** وذيل **tail** للقائمة بحيث يجب تصفيرهما قبل الاستخدام.

* لجلب خلية فارغة بمواصفات **Node** يجب استخدام تعليمة **New** .

```
node* np=New  node ;
```

* عملية قراءة او اسناد بيانة معينة لكل حقل تكون باستخدام المؤشر وكالتالي:

```
cin>>np->name;
cin>>np->avg;        }   قراءة

np->name="Ali";
np->avg=70;          }   إسناد
```

الأمثلة:

1– مشروع برنامج يوضح كيفية إنشاء وطباعة قائمة أحادية ويوضح العمليات عليها مثل الحذف (بداية ، وسط ، نهاية) و الإضافة (بداية ، نهاية ، قبل عنصر موجود ، وبعد عنصر موجود) و عملية البحث ؟

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
struct node
{
 char name[33];
 float avg;
 node *next;
 };
 void creat(node*&,node*&);
 void print(node*);
 void search(node*,node*&,node*&,char[]);
 void Delete(node*&,node*&);
 void insertFirst(node*&);
 void insertLast(node*&);
 void insertAfter(node*&,node*&);
 void insertBefore(node*&,node*&);
 void main()
 {
```

```
 clrscr();
 int ch;
 char item[33];
node* head=NULL,*tail=NULL,*cr,*pr;
  do
 {
    cout<<"Enter 1:for Creat........:"<<endl;
    cout<<"Enter 2:for print........:"<<endl;
    cout<<"Enter 3:for search.......:"<<endl;
    cout<<"Enter 4:for delete.......:"<<endl;
    cout<<"Enter 5:for Insert first.:"<<endl;
    cout<<"Enter 6:for Insert Last..:"<<endl;
    cout<<"Enter 7:for Insert After.:"<<endl;
    cout<<"Enter 8:for Insert Before:"<<endl;
    cout<<"Enter 0:to Exit..........:"<<endl;
    cin>>ch;
 if(ch==1)
  creat(head,tail);
 if(ch==2)
  print(head);
 if(ch==3)
  search(head,cr,pr,item);
 if(ch==4)
  Delete(head,tail);
 if(ch==5)
  insertFirst(head);
 if(ch==6)
  insertLast(tail);
 if(ch==7)
  insertAfter(head,tail);
 if(ch==8)
  insertBefore(head,tail);
   }
   while(ch!=0);
 }
void creat(node*&head,node*&tail)
{
 float av;
 cout<<"Enter Avg "<<endl;
 cin>>av;
 while(av>NULL)
```

```
 {
 node *np=new node;
 cout<<"Enter Name "<<endl;
 cin>>np->name;
 np->avg=av;
 if(head==NULL)
 head=np;
else
 tail->next=np;
 tail=np;
 tail->next=NULL;
 cout<<"Enter New Avg "<<endl;
 cin>>av;
 }
}
void print(node *head)
{
 while(head!=NULL)
 {
 cout<<head->name<<":" <<head->avg<<endl;
 head=head->next;
  }
 }
void search(node*head,node*&cr,node*&pr,char item[])
{
 cout<<"Enter item For search "<<endl;
 cin>>item;
 pr=0;
 cr=head;
 while((strcmp(cr->name,item)!=0)&&(cr!=0))
 {
  pr=cr;
  cr=cr->next;
  }
 if(cr!=0)
  cout<<" Found "<<endl;
 else
  cout<<" Not Found "<<endl;
  }
void Delete(node*&head,node*&tail)
{
```

```
  node*cr,*pr;
  char item[33];
  search(head,cr,pr,item);
  if(cr!=0)
  {
   if(cr==head)
    head=head->next;
    else
    if(cr==tail)
    {
     tail=pr;
     tail->next=0;
     }
    else
     pr->next=cr->next;
     delete(cr);
    }
    print(head);
   }
void insertFirst(node*&head)
 {
  float av;
  char name;
  node*np=new node;
  cout<<"Enter name"<<endl;
  cin>>np->name;
  cout<<"Enter avg"<<endl;
  cin>>av;
  np->avg=av;
  np->next=head;
  head=np;
  print(head);
 }
void insertLast(node*&tail)
 {
  float av;
  char name;
  node*np=new node;
  cout<<"Enter name"<<endl;
  cin>>np->name;
  cout<<"Enter avg"<<endl;
```

```
 cin>>av;
 np->avg=av;
 tail->next=np;
 tail=np;
 np->next=0;
}
void insertAfter(node*&head,node*&tail)
{
 float av;
 char name,item[33];
 node *cr,*pr;
 search(head,cr,pr,item);
 if(cr!=0)
 {
  if((cr!=head)&&(cr!=tail))
  {
   node*np=new node;
   cout<<"Enter name"<<endl;
   cin>>np->name;
   cout<<"Enter avg"<<endl;
   cin>>av;
   np->avg=av;
   pr->next=np;
   np->next=cr;
  }
  else
    cout<<"But ? Found in First or Last Pleese Tray agein"<<endl;
   }
  else
   cout<<" Pleese Tray agein"<<endl;
  print(head);
 }
void insertBefore(node*&head,node*&tail)
{
 float av;
 char name,item[33];
 node *cr, *pr;
 search(head,cr,pr,item);
 if(cr!=0)
 {
  if((cr!=head)&&(cr!=tail))
```

```
 {
  node*np=new node;
  cout<<"Enter name"<<endl;
  cin>>np->name;
  cout<<"Enter avg"<<endl;
  cin>>av;
  np->avg=av;
  pr=cr;
  cr=cr->next;
  pr->next=np;
  np->next=cr;
  }
 else
   cout<<"But ? Found in First or Last Pleese Tray agein"<<endl;
 }
 else
  cout<<" Pleese Tray agein"<<endl;
 print(head);
}
```

2– برنامج لانشاء قائمة احادية الارتباط كل عنصر فيها يتكون من الاسم والمعدل ، ثم انشأ قائمة جديدة تحتوي على اسماء الطلبة الناجحين من القائمة الاولى؟

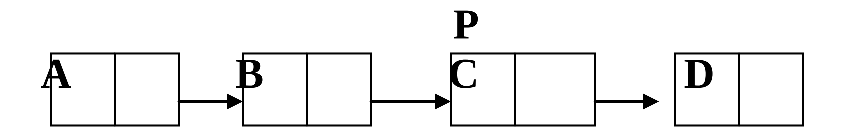
```
#include<iostream.h>
#include<string.h>
#include<conio.h>
struct node
{
 char name[33];
 float avg;
 node *next;
 };
 struct snode
{
 char name[33];
 snode *next;
 };
 void creat(node*&,node*&);
 void gen(node*,snode*&,snode*&);
 void print(snode*);
 void main()
```

```
{clrscr();
node* head=NULL,*tail=NULL;
snode* shead=NULL,*stail=NULL;
creat(head,tail);
gen(head,shead,stail);
print(shead);
}
void creat(node*&head,node*&tail)
{
 float av;
 cout<<"Enter Avg "<<endl;
 cin>>av;
 while(av>NULL)
 {
 node *np=new node;
 cout<<"Enter Name "<<endl;
 cin>>np->name;
 np->avg=av;
 if(head==NULL)
 head=np;
else
 tail->next=np;
 tail=np;
 tail->next=NULL;
 cout<<"Enter New Avg "<<endl;
 cin>>av;


 }
 }
void gen(node*head,snode*&shead,snode*&stail)
{
 while(head!=0)
 {
 if(head->avg>=50)
 {
 snode *np=new snode;
 strcpy(np->name,head->name);
 if(shead==0)
 shead=np;
else
 stail->next=np;
```

```
 stail=np;
 stail->next=0;
 }
  head=head->next;
 }
 }
void print(snode *head)
{
 while(head!=NULL)
 {
 cout<<head->name<<endl;
 head=head->next;}
  cin>>"";
 }
```

3– افرض ان لديك القائمة التالية :



اكتب الشفرة الضرورية (او اكتب برنامج) لتوضيح التالي:

أ) لحذف **P** ؟

ب) اضافة خلية الى البداية ؟

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
struct node
{
 char name[33];
 node *next;
 };
 void creat(node*&,node*&);
 void deleteP(node*&);
 void insertFirst(node*&);
 void print(node*);
 void main()
 {clrscr();
 node* head=0,*tail=0;
 creat(head,tail);
```

```
deleteP(head);
insertFirst(head);
print(head);
}
void creat(node*&head,node*&tail)
{
 char name[33];
 cout<<"Enter name "<<endl;
 cin>>name;
 while(strcmp(name,"end")!=0)
 {
 node *np=new node;
 strcpy(np->name,name);
 if(head==0)
 head=np;
else
 tail->next=np;
 tail=np;
 tail->next=0;
 cout<<"Enter New name "<<endl;
 cin>>name;
 }
 }
 void deleteP(node*&head)
 {
 node *np,*p;
 np=head->next;
 p=head->next->next;
 np->next=p->next;
 delete(p);
 }
void insertFirst(node*&head)
{
 char name;
 node*np=new node;
 cout<<"Enter name For Insert First"<<endl;
 cin>>np->name;
 np->next=head;
 head=np;
}
 void print(node *head)
```

```
{
 while(head!=0)
 {
 cout<<head->name<<endl;
 head=head->next;}
  cin>>"";
 }
```

4– برنامج لطباعة قائمة احادية الارتباط بالعكس ؟

```
#include<iostream.h>
#include<conio.h>
 struct node
 {
  int num;
  node*next;
 };
 void main()
 {
  clrscr();
  node *tail,*head,*k,*p;
  int i=1,n;
  cout<<"Enter number :"<<endl;
  cin>>n;
  node*np=new node;
  cout<<"Enter num :"<<endl;
  cin>>np->num;
  np->next=NULL;
  tail=head=np;
  while(i<n)
  {
  node*np=new node;
  cout<<"Enter new num :"<<endl;
  cin>>np->num;
  np->next=NULL;
  tail->next=np;
  i++;
  tail=np;
  }
  np=head;
```

```
k=new node;
k->num=head->num;
k->next=NULL;
p=k;
np=np->next;
while(np!=NULL)
{
k=new node;
k->num=np->num;
k->next=p;
p=k;
np=np->next;
}
np=p;
while(np!=NULL)
{
 cout<<np->num<<" ";
 np=np->next;
 }
cin>>"";
}
```

12– القوائم ثنائية الارتباط : Double Linked List

هذا النوع من القوائم يتميز باحتوائه على مؤشرين احدهما باتجاه الامام ويدعى **Next** والآخر باتجاه الخلف ويدعى **Back** بالاضافة مجموعة من الحقول البيانية ذات الانواع المختلفة .

الاعلان عنها :

head

	Next

Back	

tail

```
struct  Dnode
 {
```

```
  Data   type 1     Value1;
  Data   type 2     Value2;
  Dnode * next , *back ;
 }
```

الامثلة:

1– مشروع برنامج يوضح كيفية إنشاء وطباعة (بصورة صحيحة وبالعكس) قائمة ثنائية الارتباط ويوضح العمليات عليها مثل الحذف (بداية ، وسط ، نهاية) و الإضافة (بداية ، نهاية ، قبل عنصر موجود ، وبعد عنصر موجود) و عملية البحث ؟

```
#include<string.h>
#include<iostream.h>
#include<conio.h>
struct Dnode
{
 char name[33];
 float avg;
 Dnode *next,*back;
 };
 void creat(Dnode*&,Dnode*&);
 void print(Dnode*);
 void printinv(Dnode*);
 void search(Dnode*,Dnode*&,Dnode*&,char[]);
 void Delete(Dnode*&,Dnode*&);
 void insertFirst(Dnode*&);
 void insertLast(Dnode*&);
 void insertBefore(Dnode*&,Dnode*&);
 void insertAfter(Dnode*&,Dnode*&);
 void main()
 {
  clrscr();
  int ch;
  char item[33];
 Dnode* head=NULL,*tail=NULL,*cr,*pr;
  do
  {
    cout<<"Enter 1:for Creat........:"<<endl;
```

```
    cout<<"Enter 2:for print........:"<<endl;
    cout<<"Enter 3:for printInv.....:"<<endl;
    cout<<"Enter 4:for search.......:"<<endl;
    cout<<"Enter 5:for delete.......:"<<endl;
    cout<<"Enter 6:for Insert first.:"<<endl;
    cout<<"Enter 7:for Insert Last..:"<<endl;
    cout<<"Enter 8:for Insert Before:"<<endl;
    cout<<"Enter 9:for Insert After.:"<<endl;
    cout<<"Enter 0:to Exit..........:"<<endl;
    cin>>ch;
 if(ch==1)
  creat(head,tail);
 if(ch==2)
  print(head);
 if(ch==3)
  printinv(tail);
 if(ch==4)
  search(head,cr,pr,item);
 if(ch==5)
  Delete(head,tail);
 if(ch==6)
  insertFirst(head);
 if(ch==7)
  insertLast(tail);
 if(ch==8)
  insertBefore(head,tail);
 if(ch==9)
  insertAfter(head,tail);
   }
   while(ch!=0);
 }
void creat(Dnode*&head,Dnode*&tail)
{
 float av;
 cout<<"Enter Avg "<<endl;
 cin>>av;
 while(av>NULL)
 {
 Dnode *np=new Dnode;
 cout<<"Enter Name "<<endl;
 cin>>np->name;
```

```
 np->avg=av;
 if(head==NULL)
 {
 head=np;
 head->back=0;
 }
else
{
 tail->next=np;
 np->back=tail;
 }
 tail=np;
 tail->next=NULL;
 cout<<"Enter New Avg "<<endl;
 cin>>av;
 }
}
void print(Dnode *head)
{
 while(head!=NULL)
 {
 cout<<head->name<<": "<<head->avg<<endl;
 head=head->next;
  }
 }
void printinv(Dnode *tail)
{
 while(tail!=0)
 {
 cout<<tail->name<<": "<<tail->avg<<endl;
 tail=tail->back;
  }
 }
void search(Dnode*head,Dnode*&cr,Dnode*&pr,char item[])
{
 cout<<"Enter item For search "<<endl;
 cin>>item;
 pr=0;
 cr=head;
 while((strcmp(cr->name,item)!=0)&&(cr!=0))
 {
```

```
  pr=cr;
  cr=cr->next;
  }
 if(cr!=0)
  cout<<" Found "<<endl;
 else
  cout<<" Not Found "<<endl;
  }
void Delete(Dnode*&head,Dnode*&tail)
 {
 Dnode*cr,*pr;
 char item[33];
 search(head,cr,pr,item);
 if(cr!=0)
 {
  if(cr==head)
  {
   head=head->next;
   head->back=0;
   }
   else
   if(cr==tail)
   {
    tail=pr;
    tail->next=0;
    }
   else
    pr->next=cr->next;
    cr->next->back=pr;
    delete(cr);
   }
   print(head);
  }
void insertFirst(Dnode*&head)
 {
 float av;
 char name;
 Dnode*np=new Dnode;
 cout<<"Enter Name For Insert First"<<endl;
 cin>>np->name;
 cout<<"Enter Avg"<<endl;
```

```
  cin>>av;
  np->avg=av;
  head->back=np;
  np->next=head;
  head=np;
  np->back=0;
 }
void insertLast(Dnode*&tail)
 {
  float av;
  char name;
  Dnode*np=new Dnode;
  cout<<"Enter Name For Insert Last"<<endl;
  cin>>np->name;
  cout<<"Enter Avg"<<endl;
  cin>>av;
  np->avg=av;
  tail->next=np;
  np->back=tail;
  tail=np;
  np->next=0;
 }
void insertBefore(Dnode*&head,Dnode*&tail)
 {
  float av;
  char name,item[33];
  Dnode *cr,*pr;
  search(head,cr,pr,item);
  if(cr!=0)
  {
   if((cr!=head)&&(cr!=tail))
   {
    Dnode*np=new Dnode;
    cout<<"Enter Name For Insert After"<<endl;
    cin>>np->name;
    cout<<"Enter Avg"<<endl;
    cin>>av;
    np->avg=av;
    pr->next=np;
    np->next=cr;
    cr->back=np;
```

```
   np->back=pr;
  }
  else
    cout<<"But ? In First or Last Pleese Tray agein"<<endl;
   }
  else
   cout<<" Pleese Tray agein"<<endl;
  print(head);
 }
void insertAfter(Dnode*&head,Dnode*&tail)
 {
  float av;
  char name,item[33];
  Dnode *cr, *pr;
  search(head,cr,pr,item);
  if(cr!=0)
  {
   if((cr!=head)&&(cr!=tail))
   {
    Dnode*np=new Dnode;
    cout<<"Enter Name For Insert Before"<<endl;
    cin>>np->name;
    cout<<"Enter Avg"<<endl;
    cin>>av;
    np->avg=av;
    pr=cr;
    cr=cr->next;
    pr->next=np;
    np->next=cr;
    cr->back=np;
    np->back=pr;
    }
   else
    cout<<"But ? In First or Last Pleese Tray agein"<<endl;
   }
  else
   cout<<" Pleese Tray agein"<<endl;
  print(head);
 }
```

13- القائمة الدائرية الارتباط :

هذا النوع يحتاج الى مؤشر واحد ويرتبط العنصر الاخير مع العنصر الاول.

برنامج يوضح كيفة قراءة وطباعة قائمة دائرية الارتباط:-

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
struct cnode
{
 char name[33];
 float avg;
 cnode *next;
 };
 void creat(cnode*&,cnode*&);
 void print(cnode*,cnode*);
 void main()
 {
 clrscr();
 cnode* head=NULL,*tail=NULL;
 creat(head,tail);
 print(head,tail);
 }
 void creat(cnode*&head,cnode*&tail)
 {
  float av;
  cout<<"Enter Avg "<<endl;
  cin>>av;
  while(av>0)
  {
  cnode *np=new cnode;
  cout<<"Enter Name "<<endl;
  cin>>np->name;
  np->avg=av;
  if(head==NULL)
  head=np;
 else
  tail->next=np;
  tail=np;
  tail->next=NULL;
```

```
 cout<<"Enter Avg "<<endl;
 cin>>av;
 }
 tail->next=head;
 }
void print(cnode *head,cnode *tail)
{
 while(head!=tail)
 {
 cout<<head->name<<":" <<head->avg<<endl;
 head=head->next;
 }
 cout<<tail->name<<":" <<tail->avg<<endl;
  cin>>"";
 }
```

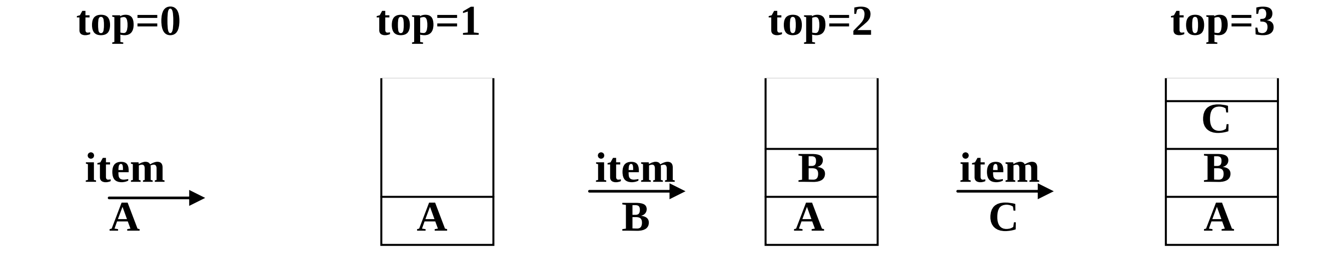
14– المكدس :Stack

هو عبارة عن هيكل بياني يتميز بكونه يشبه وعاء مفتوح من جهة الاعلى ، ويمكن التعبير عنه بمصفوفة احادية البعد او قائمة احادية الارتباط ، وكل مكدس لابد من ان يحتوي على حجم معين ، ومؤشر معين يسمى بمؤشر المكدس ويشير الى آخر عنصر موجود في المكدس ((يرمز له **Top** أو **Sp**)).

هناك عمليتين اساسيتين لادارة المكدس ((الداخل اولا يخرج آخرا)) **LIFO** :–

– عملية الدفع (**Push**) :

يزداد المؤشر بمقدار واحد على شرط كونه اقل من حجم المكدس.

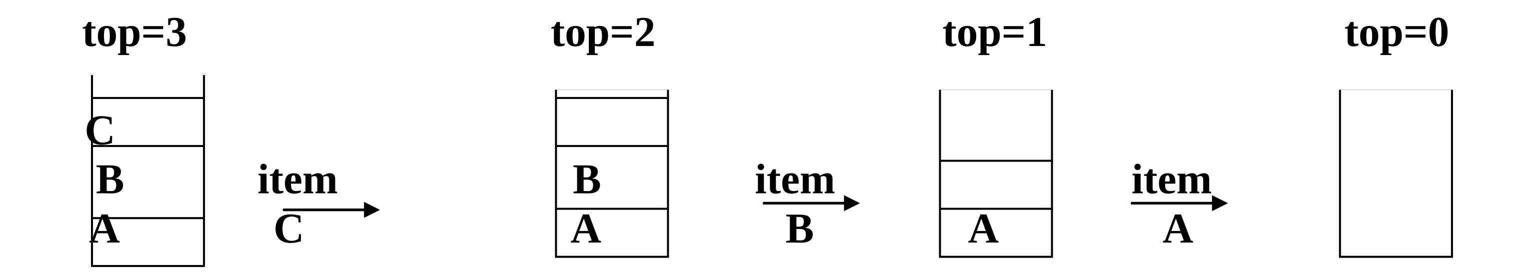


size= 3

stack is over flow المكدس ممتلئ

– عملية السحب (**Pop**) :

يتناقص المؤشر بمقدار واحد على شرط كون المكدس غير فارغ.



stack is under flow المكدس فارغ

الامثلة:

1– برنامج يوضح آليتي السحب والاضافة .

```
#include<iostream.h>
#include<conio.h>
const int size=5;
void push(int[],int&,int);
void pop(int[],int&,int&);
void main()
 {
  clrscr();
  int stack[size],top=0,item,i;
  for(i=0;i<size;i++)
   {
    cout<<"enter item"<<endl;
    cin>>item;
    push(stack,top,item);
   }
  for(i=0;i<size;i++)
   {
   pop(stack,top,item);
   cout<<item<<" ";
   }
    cin>>"";
  }
void push(int stack[size],int& top,int item)
  {
    if(top<size)
    {
     top++;
     stack[top]=item;
    }
```

```
  else
   cout<<"FULL STACK!!"<<endl;
  }
void pop(int stack[size],int& top,int& item)
 {
  if(top>0)
  {
   item=stack[top];
   top--;
  }
  else
   cout<<"EMPTY STACK??"<<endl;
  }
```

2- برنامج لانشاء مكدس بحجم 10 عناصر تمثل درجات الطلبة بمادة الهياكل ، احسب عدد الطلبة الناجحين ، ثم اطبع المكدس الاصلي ؟

```
#include<iostream.h>
#include<conio.h>
const int size=5;
void push(int[],int&,int);
void pop(int[],int&,int&);
void main()
 {
  clrscr();
  int stack[size],top=0,item,i,stack2[size],top2=0,c=0;
   for(i=0;i<size;i++)
    {
     cout<<"enter item"<<endl;
     cin>>item;
    push(stack,top,item);
    }
   for(i=0;i<size;i++)
   {
    pop(stack,top,item);
    if(item>=50)
    c++;
    push(stack2,top2,item);
   }
   for(i=0;i<size;i++)
```

```
  {
   pop(stack2,top2,item);
   push(stack,top,item);
  }
  for(i=0;i<size;i++)
  {
   pop(stack,top,item);
    cout<<item<<endl;
  }
    cout<<"The Sucess Count : "<<c<<endl;
    cin>>"";
 }
void push(int stack[size],int& top,int item)
 {
   if(top<size)
   {
    top++;
    stack[top]=item;
   }
   else
    cout<<"FULL STACK!!"<<endl;
  }
void pop(int stack[size],int& top,int& item)
 {
  if(top>0)
  {
   item=stack[top];
   top--;
  }
  else
   cout<<"EMPTY STACK??"<<endl;
  }
```

3- برنامج لانشاء مكدسين من النوع الصحيح الاول بحجم 10 عناصر والثاني بحجم 5 عناصر ، أحسب التالي:

أ) مكدس ثالث يضم العناصر الزوجية في كلا المكدسين ؟

ب) مكدس رابع يضم مضاعفات 2 في المكدس الاول ومضاعفات 3 في المكدس الثاني ؟

ج) اطبع المكدسين الجديدين ؟

```
#include<iostream.h>
#include<conio.h>
const int size=15;
void push(int[],int&,int);
void pop(int[],int&,int&);
void print(int[],int);
void main()
 {
  clrscr();
  int stack1[size],stack2[size],stack3[size],stack4[size];
  int top1=0,top2=0,top3=0,top4=0,i,item;
  for(i=0;i<10;i++)
  {
   cout<<"enter item"<<endl;
   cin>>item;
   push(stack1,top1,item);
  }
  for(i=0;i<5;i++)
  {
   cout<<"enter item"<<endl;
   cin>>item;
   push(stack2,top2,item);
  }
   for(i=0;i<10;i++)
  {
   pop(stack1,top1,item);
   if(item%2==0)
   {
    push(stack3,top3,item);
    push(stack4,top4,item);
   }}
   for(i=0;i<5;i++)
    {
     pop(stack2,top2,item);
     if(item%3==0)
     push(stack4,top4,item);
     if(item%2==0)
     push(stack3,top3,item);
    }
   print(stack3,top3);
   print(stack4,top4);
```

```
   cin>>"";
  }
void push(int stack[size],int& top,int item)
  {
   if(top<size)
   {
    top++;
    stack[top]=item;
   }
   else
    cout<<"FULL STACK!!"<<endl;
  }
void pop(int stack[size],int& top,int& item)
 {
  if(top>0)
  {
   item=stack[top];
   top--;
  }
  else
   cout<<"EMPTY STACK??"<<endl;
  }
void print(int stack[],int top)
 {
  int item;
  while(top>0)
   {
    pop(stack,top,item);
    cout<<item<<endl;
  }
 }
```

4- برنامج لانشاء مكدس بحجم 10 عناصر من النوع الصحيح ، أنشأ قائمة احادية الارتباط تحتوي على العناصر التي تتكون من مرتبتين ؟

```
#include<iostream.h>
#include<conio.h>
const int size=10;
void push(int[],int&,int);
void pop(int[],int&,int&);
struct node
```

```
 {
  int no;
  node *next;
 };
void main()
 {
  clrscr();
  int stack[size],top=0,item,i;
  node *head=0,*tail=0;
  for(i=0;i<size;i++)
   {
    cout<<"enter item"<<endl;
    cin>>item;
    push(stack,top,item);
   }
  for(i=0;i<size;i++)
   {
   pop(stack,top,item);
    if((item>9)&&(item<100))
     {
      node *np=new node;
      np->no=item;
       if(head==0)
         head=np;
       else
       tail->next=np;
       tail=np;
       tail->next=0;
      }
     }
     while(head!=0)
     {
      cout<<head->no<<endl;
      head=head->next;
   }
    cin>>"";
  }
void push(int stack[size],int& top,int item)
  {
    if(top<size)
    {
```

```
    top++;
    stack[top]=item;
   }
   else
    cout<<"FULL STACK!!"<<endl;
  }
void pop(int stack[size],int& top,int& item)
 {
  if(top>0)
  {
   item=stack[top];
   top--;
  }
  else
   cout<<"EMPTY STACK??"<<endl;
  }
```

5– برنامج لانشاء مكدس من النوع الصحيح ، ويطبع مكدس آخر يضم العناصر المكررة في المكدس الاصلي ؟

```
#include<iostream.h>
#include<conio.h>
const int size=6;
void push(int[],int&,int);
void pop(int[],int&,int&);
void print(int[],int,int);
void main()
 {
  clrscr();
  int stack[size],top=0,stack2[size],top2=0,I,j,item,a[size];
  for(i=0;i<size;i++)
   {
   cout<<"enter item"<<endl;
   cin>>item;
   push(stack,top,item);
   }
  for(i=0;i<size;i++)
   {
   pop(stack,top,item);
   a[i]=item;
   }
```

```
    for(i=0;i<size;i++)
    {
     item=a[i];
     for(j=i+1;j<size;j++)
    {
     if(item==a[j])
     push(stack2,top2,item);
    }
   }
    print(stack2,top2,item);
    cin>>"";
  }
void push(int stack[size],int& top,int item)
  {
    if(top<size)
    {
     top++;
     stack[top]=item;
    }
    else
     cout<<"FULL STACK!!"<<endl;
   }
void pop(int stack[size],int& top,int& item)
 {
   if(top>0)
   {
    item=stack[top];
    top--;
   }
   else
    cout<<"EMPTY STACK??"<<endl;
   }
void print(int stack2[size],int top2,int item)
 {
  int I;
   for(i=0;i=top2;i++)
   {
   pop(stack2,top2,item);
   cout<<item<<"   ";
   }
}
```

6- برنامج لانشاء مكدس سضم اعداد معينة ، انشأ مكدسين احدهما يضم الاعداد التامة في المكدس الاصلي ، والثاني يضم الاعداد الاولية ؟

ملحوضة: الاعداد التامة : التي تساوي مجموع مضاعفاتها.

الاعداد الاولية : التي لا تقبل القسمة الا على نفسها وعلى الواحد فقط.

```
#include<iostream.h>
#include<conio.h>
const int size=10;
void push(int[],int&,int);
void pop(int[],int&,int&);
void print(int[],int,int);
int test(int);
int pri(int);
void main()
 {
  clrscr();
  int stack[size],top=0,item,i,n,comp[size],top2=0,prim[size],top3=0;
  cout<<"Enter Size "<<endl;
  cin>>n;
  for(i=0;i<n;i++)
   {
    cout<<"enter item"<<endl;
    cin>>item;
    push(stack,top,item);
   }
    for(i=0;i<n;i++)
    {
    pop(stack,top,item);
    if((test(item))==1)
     push(comp,top2,item);
     if((pri(item))==0)
     push(prim,top3,item);
     }
     cout<<" ............. The Compelement Number "<<endl;
     print(comp,top2,item);
     cout<<" ............. The Prime Number "<<endl;
     print(prim,top3,item);
```

```
    cin>>"";
  }
void push(int stack[size],int& top,int item)
  {
   int n;
    if(top<n)
    {
     top++;
     stack[top]=item;
    }
    else
     cout<<"FULL STACK!!"<<endl;
   }
void pop(int stack[size],int& top,int& item)
 {
   if(top>0)
   {
    item=stack[top];
    top--;
   }
   else
    cout<<"EMPTY STACK??"<<endl;
   }
int test(int item)
 {
  int f=0,s=0,i;
  for(i=1;i<=item-1;i++)
  {
   if(item%i==0)
   s+=i;
   }
   if(s==item)
   f=1;
   return f;
  }
int pri(int item)
 {
  int f=0,i;
  for(i=2;i<item;i++)
  {
   if(item%i==0)
```

```
  f=1;
  }
  return f;
 }
void print(int stack[size],int top,int item)
 {
  while(top>0)
  {
   pop(stack,top,item);
   cout<<item<<endl;
   } }
```

15– فوائد (تطبيقات) المكدس :

1– ايجاد قيم التعابير الحسابية.

2– يستخدم لغايات الاستدعاء الذاتي.

3– يستخدم في عمليات الاعتراض والمقاطعة (الروتينات) في الويندوز.

4– استدعاء البرامج الفرعية.

16– التداخل الذاتي : Recartion

هو استدعاء الدالة لنفسها وفق شرط معين ، هذا الشرط يمكن ان يؤدي الى توقف الدالة (أي توقف عملية التداخل) عندما يتحقق الشرط ، او قد يؤدي الى الاسترارية حيث هنا يربرز مفهوم جديد هو مفهوم سجل التنشيط حيث يحتوي دائما على جميع المتغيرات المعلن عنها بالنهج او الظاهرة في جسم النهج ، حيث مبدأ العمل وفق القاعدة التالية:

```
if ( stop - condition)
  Msg  Box (" stop ")
else
    (continue)
{
   Reccall the same procedure which contain same process.
}
```

الامثلة:

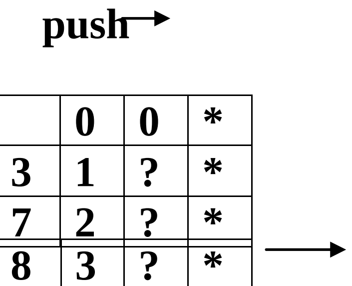
1– برنامج يستخدم مفهوم التداخل لجمع N من الاعداد ؟

توضيح بالرسم:

no	n	s	*

سجل التنشيط

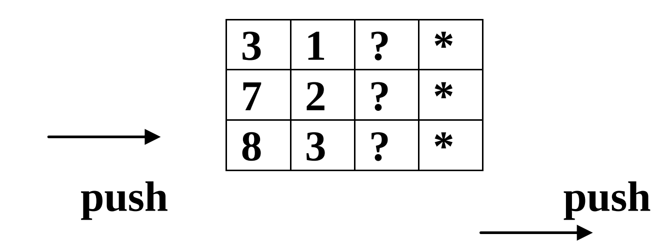
8	3	?	*



	0	0	*
3	1	?	*
7	2	?	*
8	3	?	*

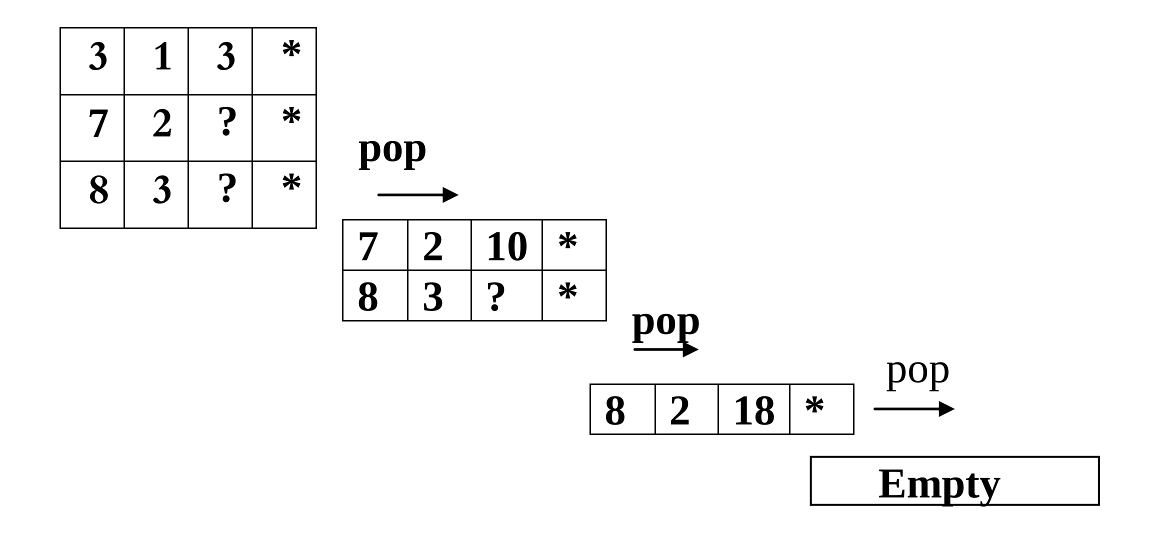
7	2	?	*
8	3	?	*

push



3	1	?	*
7	2	?	*
8	3	?	*

pop



3	1	3	*
7	2	?	*
8	3	?	*

7	2	10	*
8	3	?	*

8	2	18	*

والحل كالتالي:

```
#include<iostream.h>
#include<conio.h>
void sum(int n,int no,int& s)
 {
  if(n==0)
   s=0;
  else
    {
     cout<<"Enter No"<<endl;
   cin>>no;
    sum(n-1,no,s);
```

```
   s+=no;
   }
  }
void main()
{
 clrscr();
 int s,no,n;
  cout<<"Enter Of Size"<<endl;
  cin>>n;
  sum(n,no,s);
  cout<<s;
  cin>>""; }
```

2- برنامج يستخدم مفهوم التداخل لايجاد قيمة N! ؟

```
#include<iostream.h>
int fact(int a)
 {
  if(a==0)
  return 1;
  else
   return a*fact(a-1);
   }
void main()
{
 int f,n;
 cout<<"Enter N!"<<endl;
 cin>>n;
 f=fact(n);
 cout<<f;
 cin>>"";
 }
```

3- برنامج يستخدم مفهوم التداخل لايجاد مجموع العناصر الزوجية الموجبة في مصفوفة احادية ؟

```
#include<iostream.h>
#include<conio.h>
const int size=10;
void sum(int a[size],int n,int& s)
 {
  if(n==0)
```

```
  s=0;
 else
   {
    cout<<"Enter a[n]"<<endl;
  cin>>a[n];
   sum(a,n-1,s);
   if((a[n]%2==0)&&(a[n]>0))
   s+=a[n];
   }
  }
void main()
{
 clrscr();
 int s,b[size],m;
  cout<<"Enter m"<<endl;
  cin>>m;
  sum(b,m,s);
  cout<<s;
  cin>>"";
 }
```

4- برنامج يستخدم مفهوم التداخل لطباعة مصفوفة احادية بشكل معكوس ؟

```
#include<iostream.h>
#include<conio.h>
const int size=10;
void print(int a[size],int n)
 {
  if(n==0)
    cout<<"Print of array"<<endl;
  else
   {
    cout<<"Enter a[n]"<<endl;
  cin>>a[n];
   print(a,n-1);
   cout<<a[n]<<endl;
   }
  }
void main()
{
 clrscr();
 int a[size],n;
```

```
 cout<<"Enter Of Size"<<endl;
 cin>>n;
 print(a,n);
 cin>>"";
}
```

5– برنامج يستخدم مفهوم التداخل لحساب عدد الطلبة الناجحين في مصفوفة قيود كل عنصر من عناصرها يتكون من الاسم والمعدل والرقم الجامعي ؟

```
#include<iostream.h>
const int size=100;
struct student
 {
  char name[35];
  int avg;
 };
 void sum(student s[size],int n,int& c)
 {
  if(n==0)
   c=0;
  else
   {cout<<"Enter in formation "<<endl;
    cin>>s[n].name>>s[n].avg;
   sum(s,n-1,c);
    if(s[n].avg>=50)
    c++;}
   }
void main()
 {
  int n,c;
  student s[size];
   cout<<"Enter N "<<endl;
   cin>>n;
   sum(s,n,c);
   cout<<" the sucess: "<<c;
   cin>>" ";
  }
```

6- نهج معالجة يستخدم مفهوم التداخل لطباعة قائمة احادية الارتباط بالعكس ؟

((هذا الكود يحتاج تدقيق لانه لايطبع الهيد)).

```
void print(node *head)
 {
  if(head!=0)
  {
  head=head->next;
  print(head);
  cout<<head->No<<endl;
  }
  }
```

7- برنامج يستخدم مفهوم التداخل لحساب المقدار التالي ؟

$$\sum_{i=0}^{N} x_i^i \quad y_i^i$$

```
#include<iostream.h>
#include<math.h>
void sum(int x,int y,int n,int&s)
 {
  if(n<=0)
   s=0;
  else
   {
    sum(x,y,n-1,s);
    s=s+pow(x,n)*pow(y,n);
    }
   }
void main()
 {
  int x,y,n,s;
  cout<<"Enter N "<<endl;
  cin>>n;
  cout<<"Enter x & y "<<endl;
  cin>>x>>y;
  sum(x,y,n,s);
  cout<<s;
  cin>>"";
  }
```

8- برنامج يستخدم مفهوم التداخل لقراءة N عنصر وايجاد أكبر من بينهما ؟

```
#include<iostream.h>
```

```
#include<conio.h>
const int size=10;
void Max(int,int,int &);
void main()
 {
  clrscr();
  int no,n,max,i;
  cout<<"Enter  N  :"<<"   ";
  cin>>n;
  cout<<"--------------------"<<endl;
  Max(n,no,max);
  cout<<"\n The Max Number is: "<<max;
  cin>>" ";
 }
void Max(int n,int no,int &max)
 {
  if(n==0)
   max=0;
  else
   {
    cout<<"Enter No:- "<<"  ";
    cin>>no;
    max=no;
    Max(n-1,no,max);
    if(no>max)
    max=no;
   }
  }
```

9– برنامج يستخدم مفهوم التداخل لحساب معدل طالب معين لديه N من الدرجات ؟

```
#include<iostream.h>
#include<conio.h>
float sum(int n,int no,float& s,float av)
 {
 if(n==0)
  s=0;
 else
   {
    cout<<"Enter No"<<endl;
  cin>>no;
```

```
    sum(n-1,no,s,av);
    s+=no;
    }
    return s;
   }
void main()
 {
   clrscr();
   int no,n;
   float av,s;
   cout<<"Enter The N "<<endl;
   cin>>n;
   sum(n,no,s,av);
   av=s/n;
   cout<<"The Sumation "<<s<<endl;
   cout<<" The Average "<<av;
   cin>>"";
 }
```

10– برنامج يستخدم مفهوم التداخل لحساب المتسلسلة التالية ؟

s=1/2 + 1/4 + 2/8 + 3/16 + 5/32 + 8/64 ……… N ?

```
#include<iostream.h>
#include<math.h>
#include<conio.h>
int f,s,t;
void sum(int n,float& sn)
 {
  if(n==0)
  {
   sn=0.0;
   f=1;
   s=1;}
  else
   {
    sum(n-1,sn);
    sn+=(f/pow(2,n));
    cout<<f<<"/"<<pow(2,n)<<endl;
    t=f+s;
    f=s;
    s=t;
```

```
   }
  }
void main()
 {
  clrscr();
  int n;
  float sn;
  cout<<"Enter N "<<endl;
  cin>>n;
  sum(n,sn);
  cout<<"The Sum= "<<sn;
  cin>>"";
  }
```

11– برنامج يستخدم مفهوم التداخل لطباعة وحساب المتسلسلة التالية ؟

s= 4/1 + 5/1 + 6/2 + 7/3 ……. N ?

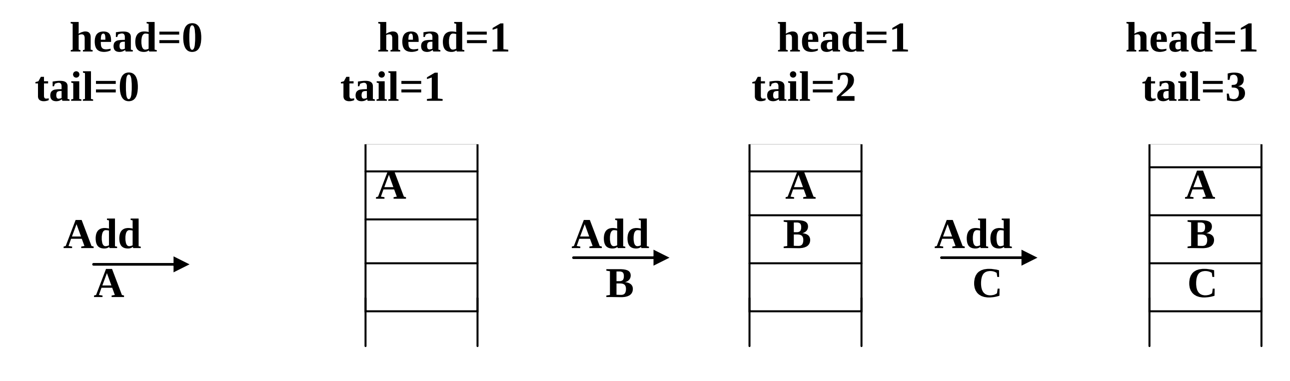
```
#include<iostream.h>
#include<conio.h>
int t,s;
float f;
void sum(int n,float& sn)
 {
  if(n==0)
   {
    sn=0.0;
    f=1;
    s=1;}
  else
   {
    sum(n-1,sn);
    sn+=((3+n)/f);
    cout<<3+n<<"/"<<f<<endl;
    t=f+s;
    f=s;
    s=t;
    }
  }
void main()
 {
  clrscr();
```

```
 int n;
 float sn;
 cout<<"Enter N "<<endl;
 cin>>n;
 sum(n,sn);
 cout<<"The Sum= "<<sn;
 cin>>"";
}
```

18– الطوابير : Queues

هو عبارة عن هيكل بياني يتميز بكونه يشبه وعاء مفتوح من الجهتين ، وكل طابور يحتوي على حجم محدد ، ومؤشرين الاول يدعى بمؤشر الرأس (**head**) ، والثاني الذيل (**tail**).

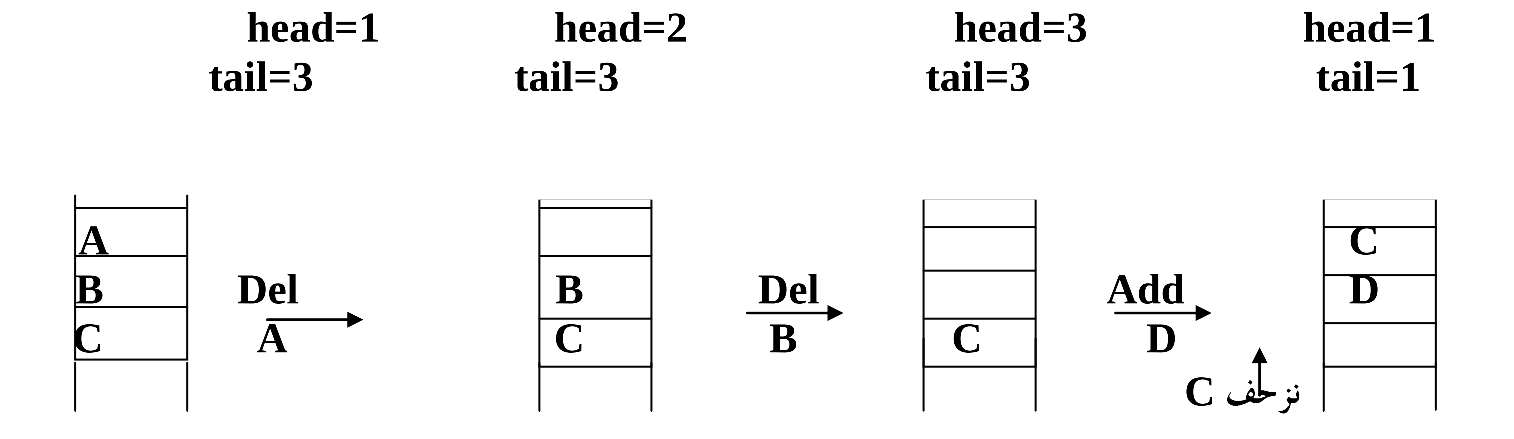
– عملية الاضافة على الطابور (**Insert**) :



size= 3

Queue is over flow الطابور ممتلئ

– عملية الحذف (**Delete**) :



head=1
tail=2

الامثلة:

1– برنامج يوضح آليتي الاضافة والحذف على الطابور :

```
#include<iostream.h>
#include<conio.h>
const int size=4;
void insert(char[],int &,int &,char);
void Delete(char[],int &h,int &,char &);
void main()
 {
  clrscr();
  int t=0,h=0;
  char q[size],item;
  for(int i=1;i<=size;i++)
  {
   cout<<"Enter item "<<endl;
   cin>>item;
   insert(q,h,t,item);
   }
   Delete(q,h,t,item);
   Delete(q,h,t,item);
  for(int j=1;j<=size;j++)
  cout<<".........."<<q[j]<<endl;
   cin>>"";
 }
void insert(char q[size],int &h,int &t,char item)
 {
  if((t-h+1)!=size)
  {
   if(h==0)
```

```
  {
   h=1;
   t=0;
   }
  else
  {
   if(t==size)
   {
    for(int i=1;i<=t-h+1;i++)
    q[i]=q[h+i-1];
    t=t-h+1;
    h=1;
    }
   }
   t++;
   q[t]=item;
   }
  else
   cout<<" Queue is Full "<<endl;
 }
void Delete(char q[size],int &h,int &t,char &item)
 {
  if(h!=0)
  {
   item=q[h];
   q[h]=0;
   if(t==h)
   h=0;
  else
   h++;
   }
  else
   cout<<" Queue is Empty "<<endl;
 }
```

2- برنامج ينشأ طابور من النوع الصحيح وبحجم 10 عناصر ، ثم ينشأ آخر يضم الاعداد الاولية في الطابور الاول ويطبع الطابور الجديد ؟

```
#include<iostream.h>
#include<conio.h>
```

```
const int size=10;
void insert(char[],int &,int &,int);
void Delete(char[],int &,int &,int &);
void print(char[],int ,int ,int);
int prim(int);
void main()
 {
  clrscr();
  int t=0,h=0,t2=0,h2=0,item;
  char q[size],q2[size];
  for(int i=1;i<=size;i++)
  {
   cout<<"Enter item "<<endl;
   cin>>item;
   insert(q,h,t,item);
   }
  for(int j=1;j<=size;j++)
  {
   Delete(q,h,t,item);
   if(prim(item)==0)
   insert(q2,h2,t2,item);
   }
   print(q2,h2,t2,item);
 }
void insert(char q[size],int &h,int &t,int item)
 {
  if((t-h+1)!=size)
  {
   if(h==0)
   {
    h=1;
    t=0;
    }
   else
   {
    if(t==size)
    {
     for(int i=1;i<=t-h+1;i++)
     q[i]=q[h+i-1];
     t=t-h+1;
     h=1;
```

```
     }
    }
    t++;
    q[t]=item;
    }
   else
    cout<<" Queue is Full "<<endl;
  }
void Delete(char q[size],int &h,int &t,int &item)
 {
  if(h!=0)
  {
   item=q[h];
   q[h]=0;
   if(t==h)
   h=0;
  else
   h++;
   }
  else
   cout<<" Queue is Empty "<<endl;
 }
int prim(int item)
 {
  int f=0,i;
  for(i=2;i<item;i++)
  {
   if(item%i==0)
   f=1;
   }
   return f;
  }
void print(char q[size],int h,int t,int item)
 {
  for(int j=1;j<t;j++)
  {
   Delete(q,h,t,item);
   cout<<".........."<<item<<endl;
   }
   cin>>"";
 }
```

3– برنامج ينشأ طابور من النوع الصحيح وبحجم 10 عناصر ، ثم ينشأ آخر يضم الاعداد التامة في الطابور الاول ويطبع الطابور الجديد ؟

```
#include<iostream.h>
#include<conio.h>
const int size=10;
void insert(char[],int &,int &,int);
void Delete(char[],int &,int &,int &);
void print(char[],int ,int ,int);
int test(int);
void main()
 {
  clrscr();
  int t=0,h=0,t2=0,h2=0,item;
  char q[size],q2[size];
  for(int i=1;i<=size;i++)
  {
   cout<<"Enter item "<<endl;
   cin>>item;
   insert(q,h,t,item);
   }
  for(int j=1;j<=size;j++)
  {
   Delete(q,h,t,item);
   if(test(item)==1)
   insert(q2,h2,t2,item);
   }
   print(q2,h2,t2,item);
 }
void insert(char q[size],int &h,int &t,int item)
 {
  if((t-h+1)!=size)
  {
   if(h==0)
   {
    h=1;
    t=0;
    }
   else
   {
```

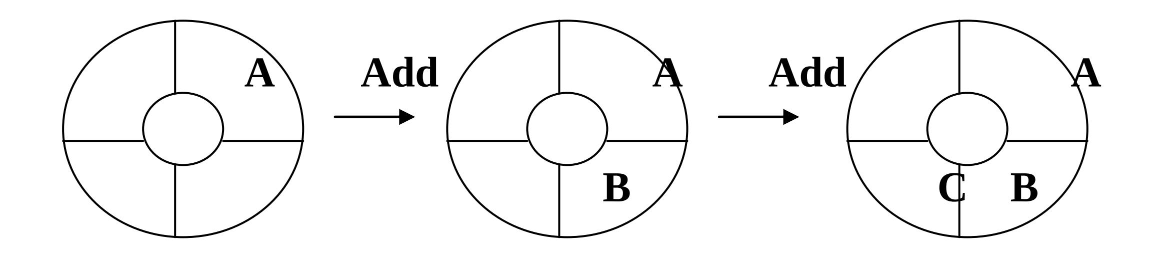
```
   if(t==size)
   {
    for(int i=1;i<=t-h+1;i++)
    q[i]=q[h+i-1];
    t=t-h+1;
    h=1;
    }
   }
   t++;
   q[t]=item;
   }
  else
   cout<<" Queue is Full "<<endl;
 }
void Delete(char q[size],int &h,int &t,int &item)
 {
  if(h!=0)
  {
   item=q[h];
   q[h]=0;
   if(t==h)
   h=0;
  else
   h++;
   }
  else
   cout<<" Queue is Empty "<<endl;
 }
int test(int item)
 {
  int s=0,i;
  for(i=1;i<item;i++)
  {
   if(item%i==0)
   s+=i;
   }
   if(s==item)
   return 1;
   return 0;
  }
void print(char q[size],int h,int t,int item)
```

```
{
 for(int j=1;j<t;j++)
 {
  Delete(q,h,t,item);
  cout<<".........."<<item<<endl;
  }
  cin>>"";
}
```

19– الطابور الدائري : Queue Circular

هو عبارة عن هيكل بياني يتكون من مجموعة من العناصر ذات حجم محدد ، بالاضافة الى احتوائه على مؤشرين الاول يدعى بمؤشر الرأس (**head**) ، والثاني الذيل (**tail**) **.**

وتطبق عليه عمليتي الاضافة والحذف كما مر سابقا ولكن بشكل دائري :



الامثلة:

1– برنامج يوضح آليتي الاضافة والحذف على الطابور الدائري :

```
#include<iostream.h>
#include<conio.h>
const int size=4;
void insert(char[],int &,int &,char);
void Delete(char[],int &,int &,char &);
void print(char[],int ,int ,char);
void main()
 {
  clrscr();
  int t=0,h=0;
  char q[size],item;
```

```
 for(int i=1;i<=size;i++)
 {
  cout<<"Enter item "<<endl;
  cin>>item;
  insert(q,h,t,item);
  }
  print(q,h,t,item);
}
void insert(char q[size],int &h,int &t,char item)
{
 if(((t-h+1)!=size)&&(t-h+1)!=0)
 {
  if(h==0)
  {
   h=1;
   t=0;
   }
  else
   if(t==size)
   t=0;
   t++;
   q[t]=item;
   }
  else
   cout<<" Queue is Full "<<endl;
 }
void Delete(char q[size],int &h,int &t,char &item)
{
 if(h!=0)
 {
  item=q[h];
  q[h]=0;
  if(h==t)
  {
  h=0;
  t=0;}
 else
  h++;
  if(h>size)
  h=1;
  }
```

```
 else
  cout<<" Queue is Empty "<<endl;
 }
void print(char q[size],int h,int t,char item)
 {
  for(int j=1;j<=size;j++)
  {
   Delete(q,h,t,item);
   cout<<".........."<<q[j]<<endl;
   }
   cin>>""; }
```

2– برنامج لانشاء طابور من نوع الرمز الواحد ، أنشأ طابور جديد يضم العناصر التي هي عبارة عن عمليات منطقية ((! ، ~ ، | ، &)) ، ثم اطبع ذلك الطابور الدائري ؟

((هذا البرنامج يحتاج الى تدقيق)).

```
#include<iostream.h>
#include<conio.h>
const int size=4;
void insert(char[],int &,int &,char);
void Delete(char[],int &,int &,char &);
void print(char[],int ,int ,char);
void main()
 {
  clrscr();
  int t=0,h=0,t2=0,h2=0;
  char q[size],q2[size],item;
  for(int i=1;i<=size;i++)
  {
   cout<<"Enter item "<<endl;
   cin>>item;
   insert(q,h,t,item);
   }
   for(i=1;i<=size;i++)
  {
   Delete(q,h,t,item);
   if((item=='&')||(item=='|')||(item=='!')||(item=='~'))
   insert(q2,h2,t2,item);
   }
   print(q2,h2,t2,item);
 }
```

```
void insert(char q[size],int &h,int &t,char item)
 {
  if(((t-h+1)!=size)&&(t-h+1)!=0)
  {
   if(h==0)
   {
    h=1;
    t=0;
    }
   else
    if(t==size)
    t=0;
    t++;
    q[t]=item;
    }
   else
    cout<<" Queue is Full "<<endl;
  }
void Delete(char q[size],int &h,int &t,char &item)
 {
  if(h!=0)
  {
   item=q[h];
   q[h]=0;
   if(h==t)
   {
   h=0;
   t=0;}
  else
   h++;
   if(h>size)
   h=1;
   }
  else
   cout<<" Queue is Empty "<<endl;
 }
void print(char q[size],int h,int t,char item)
 {
  for(int j=1;j<=size;j++)
  {
   Delete(q,h,t,item);
```

```
  cout<<"........."<<q[j]<<endl;
  }
  cin>>"";
}
```

الملفات Files

هي عبارة عن هيكل بياني يتكون من مجموعة من السطور يطلق عليها بـــ (القيود) ، كل سطر يتكون من مجموعة من الحقول ذات الانواع البيانية المختلفة والتي تمثل بطبيعة الحال أعمدة ذلك الملف ، وتقسم الملفات الى الملفات العشوائية والملفات النصية.

ملحوظة:

كل ملف ممكن ان تجري عليه مجموعة من العمليات تتمثل بالكتابة على ملف او القراءة من الملف او حذف البيانات او تعديلها ، ومن ابرز مميزات الملف كون البيانات التي تتواجد بداخله تبقى محفوظة حتى بعد انتهاء تنفيذ البرنامج وهذا ما لم نلاحظه في بقية الهياكل البيانية.

العمليات على الملفات:

الاعلان عن ملف:

يقسم الى ثلاثة اجزاء:

أ) الاعلان عن ملف لغرض الكتابة عليه:

ofstream <u>file nameP</u>("<u>path of file</u> ",mod);

الاسم البرمجي (file nameP)

مسار واسم الملف على القرص الصلب (path of file)

Ex: "c:/ali.txt"

ب) الاعلان لغرض القراءة (قراءة بيانات من ملف موجود):

ifstream file nameP("path of file ",mod);

ج) الاعلان عن الملف بشكل عام سواء للقراءة او الكتابة او للتعديل او للاضافة والحذف:

fstream file nameP("path of file ",mod);

ملحوظة:

1– كل من الدوال السابقة تتواجد ضمن مكتبة : #include<fstream.h>

2– دالة غلق الملف: file nameP.close();

3– دالة الوصول الى نهاية الملف: file nameP.eof();

الامثلة:

1– برنامج ينشأ ملف يحتوي على قيود تتألف من اسم ورقم ومعدل وبامتداد **txt** على القرص الصلب /**c:** ؟

```
#include<iostream.h>
#include<fstream.h>
void main()
{
 char name[35];
 int no;
 float avg;
 ofstream in("c:/new.data");
 cout<<"Enter avg"<<endl;  cin>>avg;
 while(avg>0)
 {
  cout<<"Enter no & name"<<endl;
  cin>>no>>name;
  in<<no<<" "<<name<<" "<<avg<<endl;
  cout<<"Enter new avg"<<endl;   cin>>avg;
  }
  in.close();
  cin>>"";
  }
```

2– برنامج يقرأ بيانات من ملف يحتوي على قيود تتألف من اسم ورقم ومعدل وبامتداد **txt** على القرص الصلب /**c:** ، ثم يحسب عدد الطلبة الناجحين ؟

```
#include<iostream.h>
#include<fstream.h>
void main()
{
 char name[35];
 int no;
 float avg;
 ifstream out("c:/new.txt");
 int c=0;
 while(out.eof()==0)
```

```
{
 out>>no>>name>>avg;
 if(avg>=50)
 c++;
 }
 cout<<c;
 cin>>"";
 out.close();
}
```

ملاحظات هامة:

اذا اردت ان تستخدم ملف على قرص صلب لابد من مراعاة الامور التالية:

1- اسم الملف الحقيقي والافتراضي (البرمجي) ونمط العمليات التي تقوم بتطبيقها على الملف.

2- التركيب والنوع البياني للحقول التي يحتويها الملف.

3- الغرض من الملف (قراءة كتابة...).

4- طريقة فتح الملف (اسم الملف والامتداد الخاص به).

5- يمكن تحديد النمط للقراءة او الكتابة او الاضافة وبالشكل التالي:

- عملية الاضافة الى نهاية الملف: **ios :: app**

Ex: fstream file("path",ios::app);

- الذهاب الى نهاية الملف عند فتحه: **ios :: ate**

- فتح الملف للقراءة فقط: **ios :: in**

- فتح الملف للكتابة عليه فقط: **ios :: out**

- فتح الملف من دون تعديل: **ios :: noreplace**

- انشاء ملف جديد (لم يوجد مسبقا) **ios :: nocreat**

- مسح محتويات ملف موجود: **ios :: trunc**

6- يمكن دمج أكثر من نمط باستخدام دالة | لكي تمكننا من فتح الملف للقراءة او للكتابة او للتعديل .. وكما موضح بالبرنامج التالي:

3- برنامج لانشاء ملف يتكون من مجموعة قيود طلبة ، كل قيد يتكون من اسم الطالب وعنوانه ومرحلته وقسمه ، ثم ينشأ ملف جديد يضم بيانات الطلبة الذين عناوينهم بصرة وفي قسم الحاسبات؟

```
#include<iostream.h>
#include<fstream.h>
```

```
#include<string.h>
void main()
{
 char name[35],add[35],stage[35],dep[35];
 fstream first("c:/student1.txt",ios::out|ios::in);
 fstream second("c:/student2.txt",ios::out);
 cout<<"Enter Name"<<endl; cin>>name;
 while(strcmp(name,"end")!=0)
 {
  cout<<"Enter Information:"<<endl;
  cin>>add>>stage>>dep;
  first<<name<<" "<<add<<" "<<stage<<" "<<dep<<endl;
  if((strcmp(add,"basra")==0)&&(strcmp(dep,"computer")==0))
  second<<name<<" "<<add<<" "<<stage<<" "<<dep<<endl;
  cout<<"Enter name:"<<endl; cin>>name;
  }
  first.close();
  second.close();
  }
```

تمت بحمد الله تعالى ما وفقنا لتدوينه مساء الاحد:

24-05-2009

علاء خضير الزيادي